DCP Builder 0.3

User's Guide

http://dsplab.diei.unipg.it/software/dcp_builder

http://www.dcpbuilder.com/

Copyright © 2009-2012 Giuseppe Baruffa, All Rights Reserved

How to contact

Websites

http://dsplab.diei.unipg.it/software/dcp_builder
http://www.dcpbuilder.com/

Technical support

Email *baruffa@diei.unipg.it, info@dcpbuilder.com*

Forum *http://dsplab.diei.unipg.it/forums/projects/dcp_builder*

Address

Digital Signal Processing Laboratory (DSPLab)
Department of Electronic and Information Engineering (DIEI)
University of Perugia
Via G. Duranti 93, I-06125 Perugia, Italy

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

First print: February 3rd, 2012

Copyright © 2009-2012 Giuseppe Baruffa, All Rights Reserved

Contents

Introduction **1**
Quick start for DCP Builder 1
What is a DCP? *1*
DCP Creation *2*
DCP delivery methods *2*
Main characteristics of DCP Builder 2
Editions *3*
Compatibility and tests *4*
Other similar programs *5*
Setup **8**
Supported platforms 8
Requirements *10*
How to download 10
Installation 11
Microsoft Windows *11*
Apple Mac OS X *12*
GNU Linux *12*
Removal 14
Microsoft Windows *14*
Apple Mac OS X *14*
GNU Linux *14*
Graphical User Interface **15**
Main panel 16
LCD section *16*
Buttons section *20*
Input panel 20
Video clips *21*
Audio clips *23*

Subtitle clips ... *23*
Clip formats ... *23*
DCDM panel ... 25
Common aspect ratios ... *28*
DCP panel ... 33
Engine panel ... 35
Trace panel ... 37
Log panel ... 38
Menu bar ... 39
File menu ... *39*
Edit menu ... *41*
View menu ... *46*
Help menu ... *48*
Status bar ... 50
How-To's ... 52
How to build a DCP ... 52
How to encode video clips ... 54
Video import in Freeware Edition ... *55*
Video import in Basic edition ... *55*
How to play a DCP without a server or projector ... 60
Command line user interface ... 61
Technical aspects overview ... 65
Technical specifications of DCPs ... 65
Picture MXF files ... *65*
Sound MXF files ... *65*
XML files ... *66*
Stereoscopic DCPs ... 66
The XYZ colorspace ... 66
How JPEG 2000 works ... 69
Lossy coding ... *69*
Sample data partitioning ... *73*
Tier-1 Lossless coding ... *74*
Tier-2 lossy coding and codestream formation ... *75*

Encoding performance of DCP Builder 76
Project file syntax 78
Comments 78
Input video section 78
InputVideoFileName [string] *78*
InputLVideoFileName [string], InputRVideoFileName [string] *78*
InputVideoFrameRate [float] *79*
InputVideoDownSample [integer] *79*
InputVideoRepeat [integer] *79*
InputVideoColourSpace [string] *79*
InputVideoCustomTransform [list of 9 floats] *79*
InputVideoGamma [float] *80*
Input audio section 80
InputAudioFileName [string] *80*
InputXAudioFileName [string] *80*
InputAudioLayout [float] *81*
Input text files 81
InputTextFileName [string] *81*
DCDM section 81
DCDMPicturePath [string] *81*
DCDMXmlPath [string] *81*
DCDMSoundPath [string] *82*
DCDMSaveType [string] *82*
DCDMPictureResolution [string] *82*
DCDMPictureResolutionLevels [string] *82*
DCDMPictureRate [integer] *82*
DCDMSoundRate [integer] *83*
DCDMPictureAspectRatio [string] *83*
DCDMPictureCustomAspectRatio [string] *83*
DCDMPictureBoxing [string] *83*
DCDMSubtitleMethod [string] *84*
DCDMJ2KPicturePath [string] *84*
DCDMNumberOfFrames [integer] *84*
DCDMFramesPerReel [integer] *84*

DCDMCueList [string] *84*
DCDMPictureBitrate [float] *85*
DCDMBitrate2 [float] *85*
DCDMBypass [string] *85*
DCP section 85
DCPPath [string] *85*
DCPType [string] *85*
DCPVersion [integer] *86*
DCPClipTitle [string] *86*
DCPTitle [string] *86*
DCPAnnotation [string] *86*
DCPIssuer [string] *86*
DCPFacility [string] *86*
DCPAgency [string] *87*
DCPLabel [string] *87*
DCPLocale [string] *87*
DCPStandard [string] *87*
Names of the DCP files [string] *88*
Cryptography section (EXPERIMENTAL - DON'T USE) 88
CryptoEncrypt [boolean] *88*
Compression engine section 89
EngineNumberOfThreads [integer] *89*
EnginePictureResize [string] *89*
EngineSoundResample [string] *89*
EngineResume [string] *89*
EngineTraceFileName [string] *90*
EngineTraceContents [list of strings] *90*
EngineLogFileName [string] *90*
EngineLog [list of strings] *90*
Registration 91
How to obtain a registration code 92
Why I ask for your machine fingerprint? 92
Frequently Asked Questions (FAQ) 93
License 96

DCP Builder 96
Included libraries 97
Zlib *97*
Libbzip2 ..... *97*
LibTIFF *98*
Libav *99*
LibPNG *99*
Libnsbmp *99*
LibFLAC *99*
LibTarga *99*
Libxml2 ..... *99*
OpenDPX *99*
OpenJPEG *100*
SQLCipher *100*
XMLSec *100*
Bitstream Vera Sans *100*
Glossary **102**
Bibliography **104**
Credits **106**
People 106
Auxiliary libraries, software, and other things 107
Changelog **109**
Document history **111**

Introduction

Quick start for DCP Builder

DCP Builder is a natively multiplatform software that can be used to create Digital Cinema Packages (DCPs), which can be thus played on Digital Cinema projectors.

- If you are an expert and just need a quick start for creating a DCP using DCP Builder, skip to the section "How to build a DCP".
- If you want to understand well the operation of the program, you need to read first the "Graphical User Interface" section and then the "Project file syntax" section.
- If you are still more curious about how a DCP is done "internally", explore the section "Technical specifications of DCPs".

What is a DCP?

A Digital Cinema Package is a collection of digital files used to store and convey Digital Cinema (DC) audio, image, and data streams.[1]

The term has been defined by Digital Cinema Initiatives, LLC in their recommendations for packaging of DC contents.[2] Following to the recommendations, general practice adopts a file structure that is organized into a number of (generally) multi-gigabyte size *Material eXchange Format* (MXF) files, which are separately used to store audio and video streams, and auxiliary index files in XML format. One such file, the Composition Playlist, defines the playback order of Digital Cinema Packages during presentation.

The MXF files contain streams that are compressed, encoded, and encrypted, in order to reduce the huge amount of required storage and to protect from unauthorized use. The image part is JPEG 2000 compressed, whereas the audio part is linear PCM. The adopted (optional) encryption standard is AES 128 bit in CBC mode.

SMPTE standards are used to conform to the recommendations among different tool vendors and producers.

---

[1] Digital Cinema Package. (n.d.). In *Wikipedia*. Retrieved May 10, 2011, from http://en.wikipedia.org/wiki/Digital_Cinema_Package.

[2] http://www.dcimovies.com/.

For more technical information on DCPs, see the “Technical aspects” section.

DCP Creation

Film producers and distributors generally rely on digital cinema encoding facilities to produce and quality control check a digital cinema package before release. Facilities follow guidelines set out in the recommendations to ensure compatibility with all digital cinema equipment. For bigger studio release films, the facility will usually create a DCDM (Digital Cinema Distribution Master).

A DCDM is similar to DCP, only the frames are in either DPX or TIFF format and both sound and picture are not yet wrapped into MXF files. The frames are then compressed with the JPEG 2000 algorithm, and this step is denoted as DCDM*.

A DCP can be encoded directly from a DCDM. A DCDM is useful for archiving purposes and also facilities can share them for international re-versioning purposes. They can easily be turned into alternative version DCPs for foreign territories. For smaller release films, the facility will usually skip the creation of a DCDM and instead encode directly from the DSM (Digital Source Master) the original film supplied to the encoding facility. A DSM can be supplied in a multitude of formats and color spaces. For this reason, the encoding facility need to have extensive knowledge in color space handling including, on occasion, the use of 3D LUTs to carefully match the look of the finished DCP to a celluloid film print. This can be a highly involved process in which the DCP and the film print are "butterflied" (shown side by side) in a highly calibrated cinema.

Less demanding DCPs are encoded from tape formats such as HDCAM SR. Quality control checks are always performed in calibrated cinemas and carefully checked for errors. QC checks are often attended by colorists, directors, sound mixers, and other personnel to check for correct picture and sound reproduction in the finished DCP.

DCP delivery methods

The most common method uses a hard disk designed specifically for digital cinema servers to ingest from. The hard drives are usually formatted in the Linux EXT2 or EXT3 format. NTFS is occasionally used but is generally not considered reliable as some Linux based digital cinema servers are unable to read this format. Hard drive units are normally hired from a digital cinema encoding company, sometimes in quantities of thousands. Drives are commonly shipped in protective hard cases. The drives are delivered via Express Courier to the exhibition site. Other, less common methods adopt a full digital delivery, using either dedicated satellite links or high speed Internet connections.

Main characteristics of DCP Builder

Main features of the program are:

- multiplatform - Microsoft Windows, Mac OS X, Linux, 32 and 64 bit;
- 2K, 4K, and 3D picture format;
- Flat, scope, HDTV, full, and other custom aspect ratios;
- 24, 25, 30, 48, 50, and 60 FPS frame rates are supported;
- 48/96 kHz sound sampling rate;
- 1.0, 2.0, 5.1, 7.1, and (up to) 16 audio channels layout;
- converts audio clip formats (WAV, FLAC) plus TIFF/PNG/TGA/DPX/BMP sequence of pictures. It also converts nearly all video formats, if the final user wants to;
- supports multiple origin RGB color spaces (PAL, NTSC, BT.709, sRGB, etc.);
- can save Digital Cinema Distribution Master (DCDM and DCDM*) in PPM/AU/J2K/XML format;
- can use different files for audio, video, and text input;
- supports SMPTE and *Interop* DCP formats;
- can resume work from partially completed DCDM or DCDM*;
- subtitles in burn-in and *Interop* mode;
- DCP finalization success email.

What it does not support:

- MPEG-2 DCPs;
- assembly of multiple input clips;
- video frame rate resampling (audio is slowed-down or speeded-up);
- pitch shift of audio;
- encryption, signing, and generation of Key Delivery Message (KDM).

Editions

DCP Builder is customized in several editions, in order to satisfy the requirements of the different types of users. Thus, in this user's guide, not all options will be available to all versions of DCP Builder. This will be signaled with a proper logo, e.g., AVAILABLE ONLY IN .... Table 1 lists and compares all the salient characteristics of the different editions: please note that, at the moment of writing, not all editions could be available, and their features could change without notice.

Table 1 - Overview of DCP Builder editions and their features.

	FE **Freeware Edition**	BE **Basic Edition**	AE **Advanced Edition**
Microsoft Windows version	√	√	√
Apple Mac OS X version	—	√	√
GNU Linux version	—	√	√
2K & 3D	√	√	√
4K	—	√	√
16 channels audio layout	—	√	√
Subtitles	—	√	√
Imports BMP, TIFF, DPX, PNG, TGA picture sequences	√	√	√
Imports WAV, FLAC audio clips	√	√	√
Imports video clips[3]	—	√	√
Hyperthreading limit	4	16	24
Command line version	—	—	√
Sends email when done	—	√	√
Custom aspect ratio	—	√	√
Custom colorspace	—	√	√
C Quick JPEG 2000 compression engine	—	—	√
Digital signature of XML files	—	—	√
Reels encryption	—	—	√

Compatibility and tests

The DCPs prepared by this program have been tested on the following DC servers or DC software:

- Hardware players
    - Doremi Labs DCP-2000;[4]

[3] Video clip import is available only after user action.

  - Kodak JMN-3000;
  - Kodak SMS;
  - Dolby dss100;[5]
  - Dolby dss200;[6]
  - Doremi Labs IMB-4K;[7]
  - XDC CineStore Solo G3;[8]
- Software players
  - Fraunhofer's *EasyDCP player*;[9]
  - digitAll's *dcpPlayer*;[10]
  - 3dtv.at's *Stereoscopic player*.[11]

Other similar programs

Table 2 presents a list of programs that, to my (and Wikipedia) knowledge, are able to prepare DCPs. For further details, please check out Wikipedia page.[12]

Table 2 - DCP encoding software at a glance.

Name	**JPEG 2000**	**XYZ**	**MXF**	**XML**	**Frame size**	**3D**	**OS**	**License**
2DCP_GUI[13]	Yes	Yes	Yes	Yes	2K	Yes	Windows	Freeware
CineCert AS-DCP File	No	No	Yes	No	2K, 4K	Yes	Windows, Mac	BSD-like

[4] http://www.doremicinema.com/dcp.html

[5] http://www.dolby.com/DocLibProductLanding.aspx?taxid=158

[6] http://www.dolby.com/professional/products/cinema/digital-cinema/dss200.html

[7] http://www.doremicinema.com/PDF/ShowIMB.pdf

[8] http://www.xdcinema.com/

[9] http://www.dcinema.fraunhofer.de/en/veranstaltungen/ibc/player.jsp

[10] http://digitall.net.au/products/dcp-player.html

[11] http://www.3dtv.at/Movies/Index_en.aspx

[12] http://en.wikipedia.org/wiki/Digital_Cinema_Package

[13] http://www.mik-digital.de/programme

Name	JPEG 2000	XYZ	MXF	XML	Frame size	3D	OS	License
Access Library[14]							OS X, Linux	
Doremi CineAsset[15]	Yes	Yes	Yes	Yes	2K, 4K		Windows, Mac OS X	Proprietary
Cinemaslides[16]	Yes	Yes	Yes	Yes	2K, 4K	No		
DVS Clipster[17]	Yes	Yes	Yes	Yes	2K, 4K		Windows	Proprietary
DCPC - Digital Cinema Package Creator[18]	Yes	Yes	Yes	Yes	2K, 4K	Yes	Windows	
Prism DCP Encoder[19]	Yes	Yes	Yes	Yes	2K, 4K	Yes	Windows	Proprietary
Yade DCP Encoder	Yes	Yes	Yes	Yes	2K, 4K	Yes	Windows, Linux	Proprietary
Inition DCP Pro[20]	Yes	Yes	Yes	Yes	2K	Yes	Windows	Proprietary
DCP Tool[21]	Yes	Yes	Yes	Yes	2K, 4K	Yes	Windows	Proprietary
Dietrich[22]	Yes	Yes	Yes	Yes	2K, 4K	Yes	Linux	
Fraunhofer IIS easyDCP Creator[23]	Yes	Yes	Yes	Yes	2K, 4K	Yes	Windows, Mac OS X	Proprietary
MagnaMana FinalDCP[24]	Yes	Yes	Yes	Yes	2K, 4K		Windows, Mac OS X	Proprietary
Extron J2KENC[25]								

[14] http://www.cinecert.com/asdcplib/

[15] http://www.doremicinema.com/cineasset.html

[16] https://github.com/wolfgangw/digital_cinema_tools/wiki/Cinemaslides

[17] http://www.dvs.de/products/video-systems/clipster.html

[18] http://cinema.terminal-entry.de/

[19] http://www.prismdcp.com/

[20] http://www.inition.co.uk/inition/dispatcher.php?action=get&model=products&URL_=product_stereovis_dcppro

[21] http://www.hs-rm.de/ing/ueber-uns/personen/personen-im-fb-ing/prof-dr-ing-wolfgang-ruppel/mastering-workflow/index.html

[22] https://github.com/wolfgangw/dietrich/wiki

[23] http://www.iis.fraunhofer.de/EN/bf/bv/cinema/dcpcreation.jsp

[24] http://www.magnamana.com/products/finaldcp/

[25] http://www.extron.com/product/product.aspx?id=j2kenc&s=5

Name	JPEG 2000	XYZ	MXF	XML	Frame size	3D	OS	License
opencinematools[26]	Yes	No	Yes	Yes	2K, 4K	Yes	Windows, Mac OS X, Linux	BSD
OpenCubeDCP[27]	Yes	Yes	Yes	Yes	2K, 4K	Yes	Linux	Proprietary
OpenDCP[28]	Yes	Yes	Yes	Yes	2K, 4K	Yes	Windows, Linux, Mac OS X	GPL
Qube QubeMaster Pro[29]	Yes	Yes	Yes	Yes	2K, 4K		Windows	Proprietary
Dolby SCC2000[30]	Yes	Yes	Yes	Yes	2K, 4K	Yes	Linux	Proprietary
Cinecert Wailua[31]	Yes	Yes	Yes	Yes	2K, 4K	Yes	Mac OS X, Linux	Proprietary
QuVIS Wraptor[32]	Yes	Yes	Yes	Yes	2K	No	Mac OS X	Proprietary

[26] http://code.google.com/p/opencinematools/

[27] http://www.opencubetech.com/page81/OpenCubeDCP

[28] http://code.google.com/p/opendcp/

[29] http://www.qubecinema.com/qubemaster.htm

[30] http://www.dolby.com/professional/products/cinema/digital-cinema/scc2000.html

[31] http://www.cinecert.com/products/

[32] http://www.quvis.com/?Action=Products&SubAction=wraptor

Setup

Supported platforms

DCP Builder is natively multiplatform: it has been written in C and C++, uses the wxWidgets GUI library[33] and it can be ported to multiple OS and CPU architectures.

The currently supported platforms are:

- Microsoft Windows, 32 and 64 bit (XP/Vista/7);
- Mac OS X Intel, 32 and 64 bit (Leopard, Snow Leopard, and following versions);
- GNU Linux, 32 and 64 bit (Ubuntu/openSUSE and possibly other compatible distributions);[34]
- FreeBSD (experimental).

In the following, if not differently specified, we will always refer to the Windows version for screenshots and menus. Linux and Mac OS X versions follow exactly the same GUI paradigm, though with minor differences to respect OS interface (minimize/close buttons position, menu position, etc.).

[33] http://www.wxwidgets.org/

[34] Ubuntu Hardy Heron 32 bit and higher, Ubuntu Lucid Lynx 64 bit and higher. openSUSE 11.4 32 bit and higher, openSUSE 12.1 64 bit and higher.

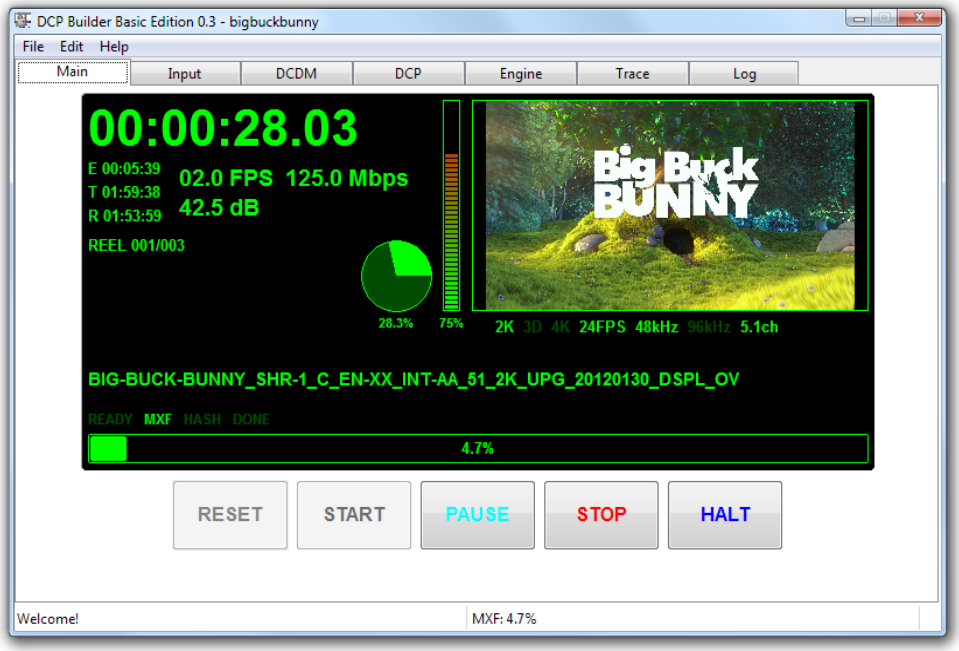


Figure 1 - DCP Builder with Aero interface running on Windows 7.

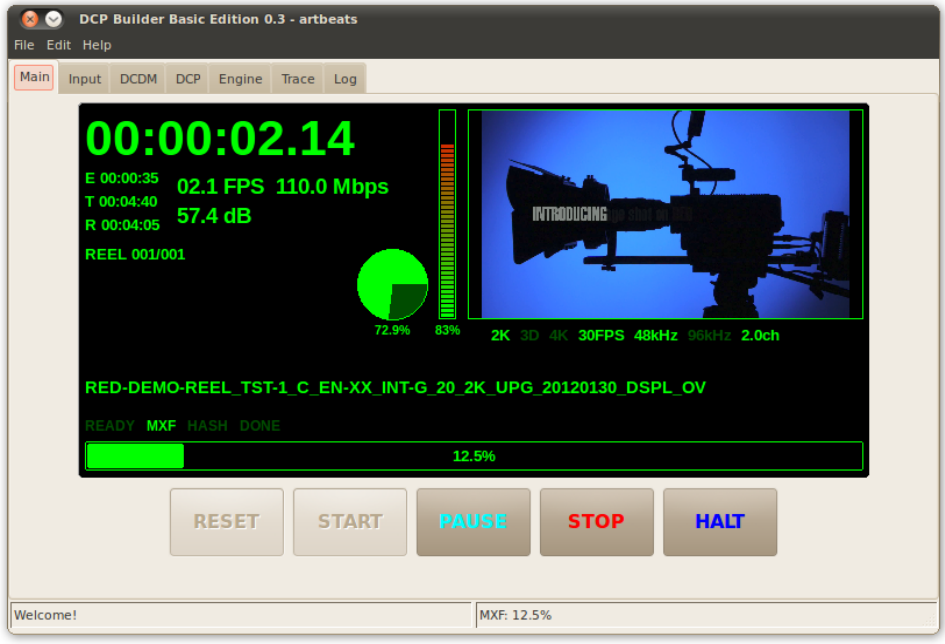


Figure 2 - DCP Builder with GTK2 interface running on Ubuntu Linux.

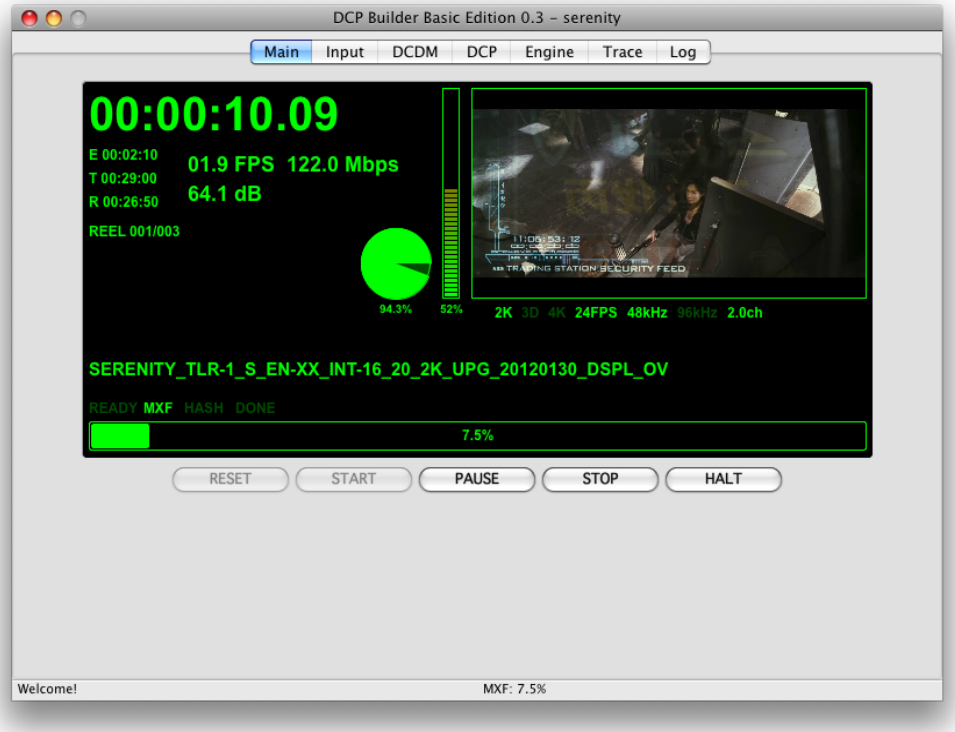


Figure 3 - DCP Builder with Quartz interface running on Mac OS X Leopard.

Requirements

- 1 GHz CPU or higher (2 GHz multi-core CPU is preferred);
- 1 GB RAM or higher (2 GB is preferred);
- 500 GB free disk space available or higher, for clips of duration of 2 hours.

How to download

You can download DCP Builder only from the following websites (the DCPBUILDER.COM domain is generally updated less often)

http://dsplab.diei.unipg.it/software/dcp_builder

http://www.dcpbuilder.com/

The direct download link is

http://dante.diei.unipg.it/~baruffa/DCPBuilder/

http://www.dcpbuilder.com/download

The download folder is organized as:

- dcp/
  - The folder may contain some short testing DCPs built with DCP Builder.
- docs/
  - This folder contains general documentation and the PDF user manual.
- linux/
  - This folder contains 32 and 64 bit installation packages for GNU Linux.
- macosx/
  - This folder contains 32 and 64 bit disk images for Mac OS X.
- windows/
  - This folder contains the 32 and 64 bit installers for Microsoft Windows.

Installation

You must download the version that is most suited to your hardware and OS: please note that 32 bit versions can be installed on 32 and 64 bit platforms, generally, but they do not necessarily provide the same degrees of freedom.

Please be careful when installing 32 and 64 bit versions at the same time, alongside operation has not been tested.

Microsoft Windows

Run the *.exe* installer (fig. 4) and follow on-screen instructions.

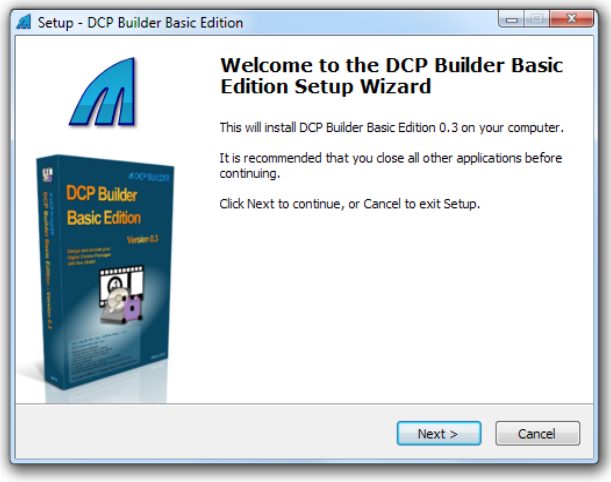

Setup - DCP Builder Basic Edition
Welcome to the DCP Builder Basic Edition Setup Wizard
This will install DCP Builder Basic Edition 0.3 on your computer.
It is recommended that you close all other applications before continuing.
Click Next to continue, or Cancel to exit Setup.
DCP Builder Basic Edition
Next >
Cancel

Figure 4 - DCP Builder setup running under Microsoft Windows.

Apple Mac OS X

Double click on the *.dmg* disk image and drag-and-drop the program over the ***Applications*** folder (fig. 5). You may omit to copy the manual, since it is a link to the manual embedded in the bundle.

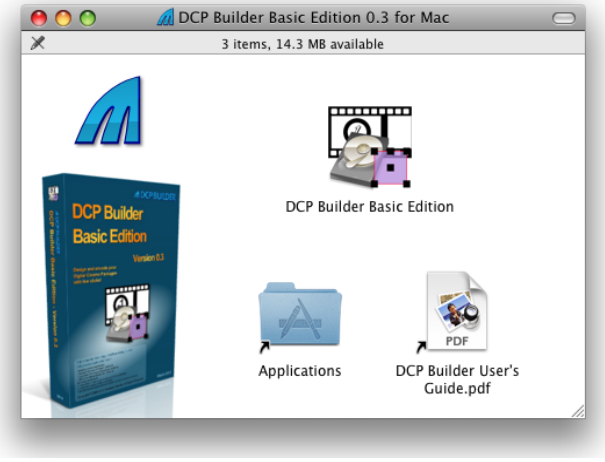


Figure 5 - DCP Builder disk image contents under Mac OS X Leopard.

GNU Linux

Double click on the *.deb* or *.rpm* archive and install with programs like *gdebi* or *KPackageKit* (figs. 6 and 7).

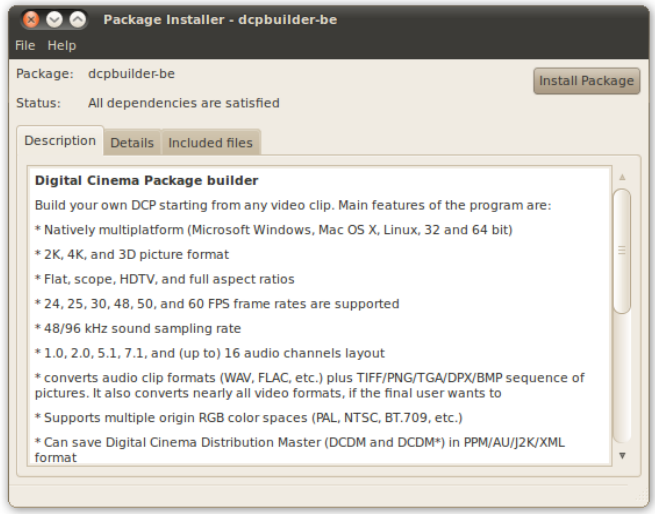

Package Installer - dcpbuilder-be
File Help
Package: dcpbuilder-be
Install Package
Status: All dependencies are satisfied
Description Details Included files
Digital Cinema Package builder
Build your own DCP starting from any video clip. Main features of the program are:
* Natively multiplatform (Microsoft Windows, Mac OS X, Linux, 32 and 64 bit)
* 2K, 4K, and 3D picture format
* Flat, scope, HDTV, and full aspect ratios
* 24, 25, 30, 48, 50, and 60 FPS frame rates are supported
* 48/96 kHz sound sampling rate
* 1.0, 2.0, 5.1, 7.1, and (up to) 16 audio channels layout
* converts audio clip formats (WAV, FLAC, etc.) plus TIFF/PNG/TGA/DPX/BMP sequence of pictures. It also converts nearly all video formats, if the final user wants to
* Supports multiple origin RGB color spaces (PAL, NTSC, BT.709, etc.)
* Can save Digital Cinema Distribution Master (DCDM and DCDM*) in PPM/AU/J2K/XML format

Figure 6 - DCP Builder setup running within 'gdebi' under Ubuntu Linux.

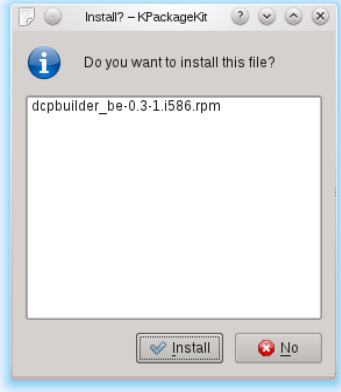


Figure 7 - DCP Builder setup running within 'KPackageKit' under openSUSE Linux.

Differently, you can install it from the command line with (Ubuntu)

```
#> sudo dpkg -i dcpbuilder_03_be_i386.deb
```

or (openSUSE)

```
#> sudo zypper install dcpbuilder_be-0.3-1.i586.rpm
```

Removal

Microsoft Windows

Uninstall the program from the Control Panel or from the proper program menu entry, and follow the given instructions.

Apple Mac OS X

Delete the application from the ***Applications*** folder.

GNU Linux

Uninstall the package using *Ubuntu Software Center*, *synaptics*, or *yast*. From the command line, you can issue the following commands

```
#> sudo dpkg -r dcpbuilder
```

or

```
#> sudo apt-get remove dcpbuilder
```

or

```
#> sudo rpm -e dcpbuilder
```

or

```
#> sudo zypper remove dcpbuilder
```

Graphical User Interface

The GUI has two different operating modes, compact and extended, which can be selected in the program preferences. The program needs to be restarted in order to change interface.

In compact mode, there is only one window (see previous figures), which you can manage via multiple tabs.

In expanded mode (fig. 8), there are many windows, which can be positioned on the desktop as preferred (if you have two monitors, you could use one of them to see a real color full screen preview of the encoding results).

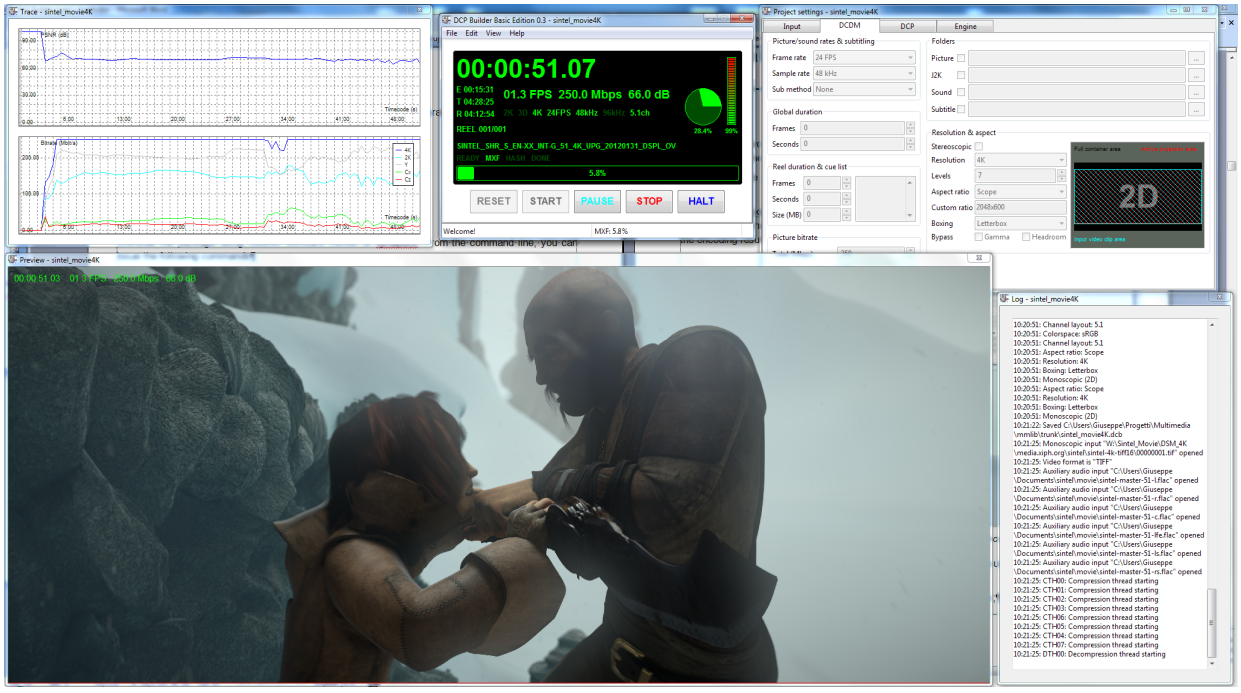


Figure 8 - DCP Builder running in expanded mode.

In both cases, a number of panels are, each one, dedicated to a particular task or parameters set. They are:

- ***Main*** panel, hosting the LCD display and the operating buttons;

- ***Input*** panel, for the selection of input audio/image clips and their relevant parameters;
- ***DCDM*** panel: to specify the format of the DCP;
- ***DCP*** panel, to set the compression parameters and DCP metadata;
- ***Engine*** panel, to adjust compression settings;
- ***Trace*** panel, hosting a timed log of encoding quality.
- ***Log*** panel, hosting a timed log of actions and responses.

Main panel

The main panel (fig. 9) is composed by two sections: the LCD section and the buttons section.

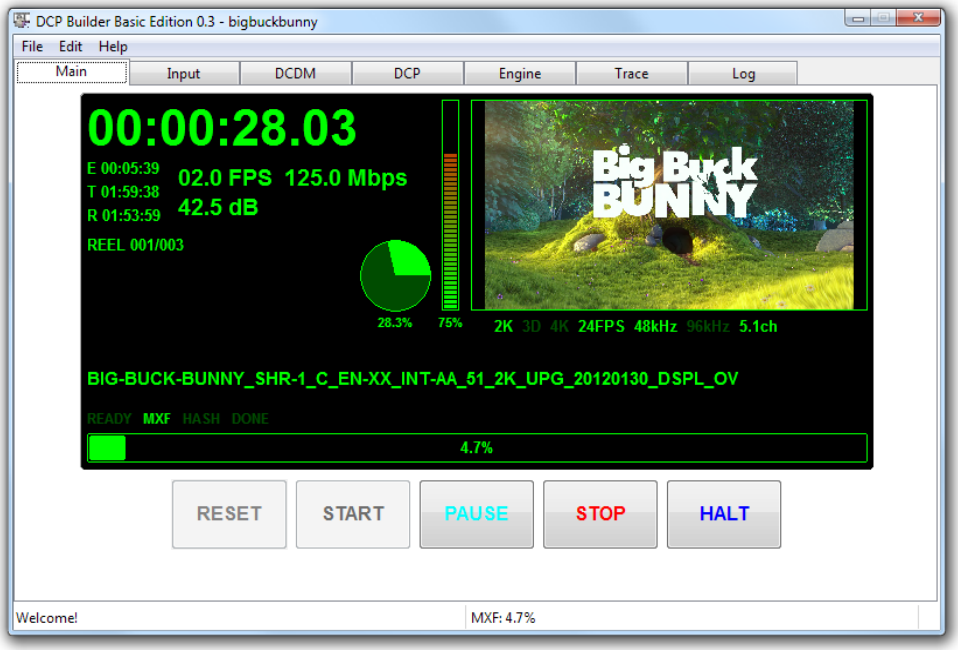


Figure 9 - The main panel, with the LCD and the operating buttons.

LCD section

The LCD shows some basic information that keeps you up-to-date with the current and actual processing:

- Time code: in HH:MM:SS.FF format, represents the current advancement status of the DCP (fig. 10).

Figure 10 - Time code.

- Elapsed time (E): in HH:MM:SS format, shows the elapsed time since encoding began (fig. 11, top).
- Total time (T): in HH:MM:SS format, shows the expected total time for the encoding (fig. 11, middle).
- Remaining time (R): in HH:MM:SS format, shows the expected remaining time for the encoding to finish (fig. 11, bottom).

Figure 11 - Elapsed (E), total (T), and remaining (R) processing time.

- Reel number: in CURRENT/TOTAL format, shows the current reel being worked on and the total number of reels that will be done (fig. 12).

Figure 12 - Current and total number of reels.

- Frame rate: given in FPS, is a direct indicator of the speed of your CPU in fulfilling the encoding task (fig. 13, top left).
- Bit rate: given in Mbps, shows how much your current frame is actually being compressed (fig. 13, top right).
- PSNR quality: given in dB, shows how much your current frame is being corrupted from compression artifacts (fig. 13, bottom).

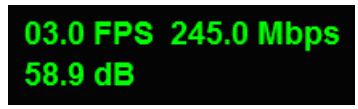


Figure 13 - Processing frame rate, actual compressed bit rate, and PSNR quality.

- Machine load: disk usage is represented in terms of percentage of occupied disk space (the disk is that where the DCP is being saved), whereas CPU load is expressed as percent of time dedicated to the encoding process (fig. 14).

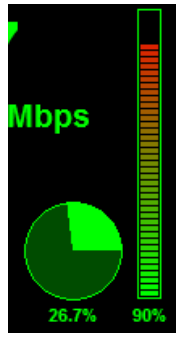


Figure 14 – Disk usage and CPU load.

- Preview window: shows a frame every *n* frames, after decoding. It allows you to predict if the DCP is being done without problems and allows you to keep an eye to the advancement status. Preview can be enabled/disabled via the ***Edit→Preview*** menu (fig. 15, top).

> NOTE: PLEASE DO NOT RELY ON DCP BUILDER PREVIEW FOR TONE CORRECTIONS!
>
> NOTE: REAL-TIME PREVIEW IS A COSTLY OPERATION, SINCE YOUR JUST-ENCODED FRAMES MUST BE COMPLETELY DECODED. SET A PROPER UPDATE PERIOD OR DISABLE THE PREVIEW COMPLETELY IF YOUR MAIN CONCERN IS SPEED.

- DCP formats: line summarizing the main characteristics of the DCP being created (fig. 15, bottom).

Figure 15 - Preview window and DCP characteristics one-line summary.

- DCP title (fig. 16): this is imported from the DCDM panel.

Figure 16 - DCP title.

- Advancement status (fig. 17, top left):
  - **READY** when the program is ready to encode;
  - **MXF** during encoding;
  - **HASH** during the final hashing phase;
  - blinking **DONE** when the DCP is ready.
- Advancement slider: shows the percentage of MXF done, during **MXF** status, or of hashing done, during **HASH** status (fig. 17, bottom).

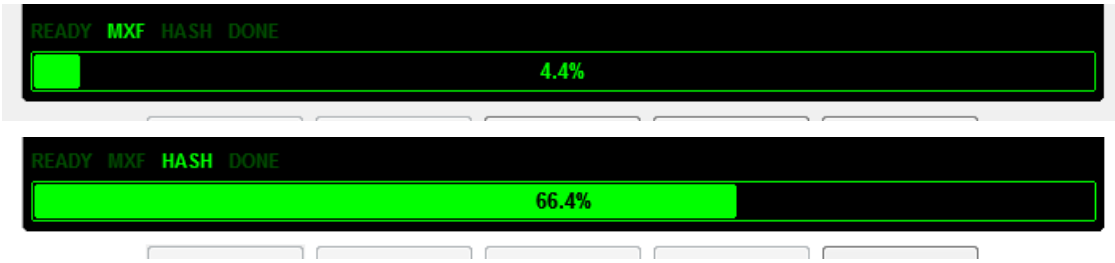


Figure 17 - Advancement status and percentage slider during MXF (top) and HASH (bottom) stages.

You can drag-and-drop project files (with *.dcb* extension) over the LCD for opening them.

Buttons section

There are five buttons that allow controlling the operation:

- **RESET**: after a stop or halt action, resets the encoding status.
- **START**: begins the encoding process.
- **PAUSE**: pause the encoding process (note: this may not be implemented in current release).
- **STOP**: gracefully stop the encoding process, meaning that a DCP will be created up to the currently processed frame.
- **HALT**: aborts DCP creation entirely, resulting DCP is unplayable.

Figure 18 - Operation buttons.

Input panel

The input panel (fig. 19) allows the user to enter the audio and video clips to be converted into the DCP. Such files can be dragged-and-dropped from your file manager onto the Video, Audio, and Text input panels.[35]

[35] It does not work in the Windows version.

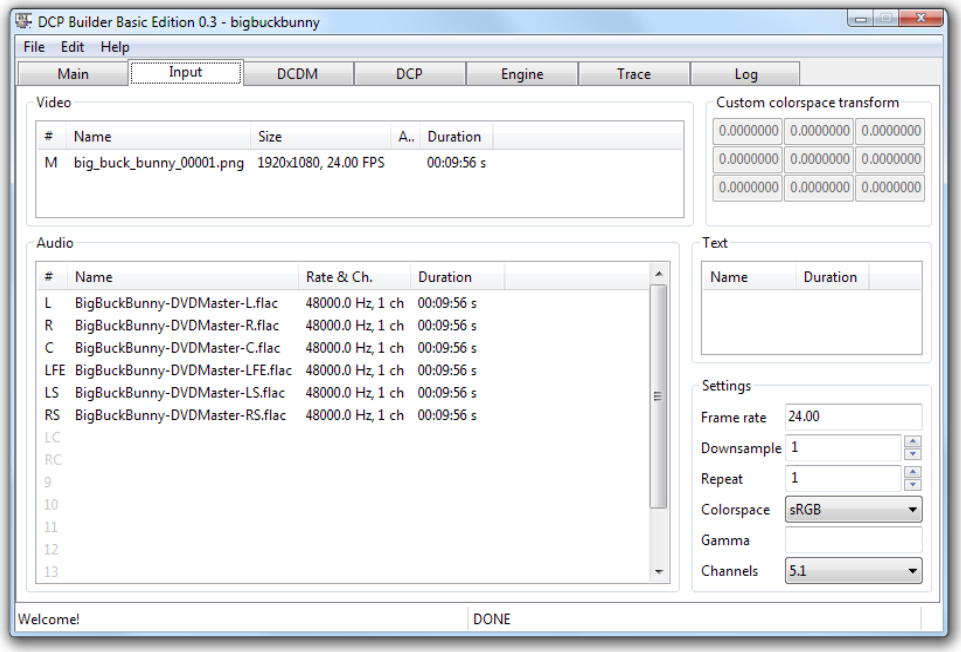


Figure 19 - The input panel.

Video clips

Enter video clips by right-clicking on the list box (fig. 20) and selecting the files you want to include in your project. The video clip must be in one of the supported formats (see tab. 4): in case of a sequence of images, select the first image (the one with the lowest ordinal).

Similarly, files can be also removed from the lists. Use **+** and **–** keys to raise up/lower down a file in the list.

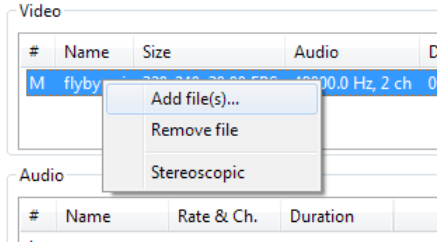


Figure 20 - Pop-up menu in input clip panel.

You can also decide to enter two different files, one per view, when encoding stereoscopic DCPs (fig. 21).

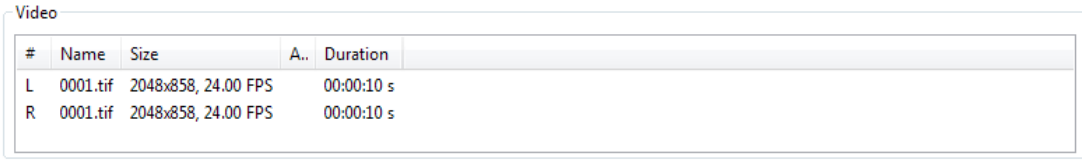


Figure 21 - Stereoscopic setup in input clip panel.

You cannot enter more than two files: this program does not do nonlinear editing operations such as cut-and-paste of video clips. You must have your final clip already prepared.

Adjust the ***Frame rate***, ***Downsample***, and ***Repeat*** values when encoding from a list of sequentially numbered images. Thus, BEFORE entering the sequence of images clip in the video input panel:

- select the proper frame rate (a sequence of images has, generally, no means to convey frame rate information);
- choose how many images you want to take out of every group of *n* (generally, this is set to 1);
- choose how many times you want to cyclically repeat that sequence (generally, this is set to 1). For instance, you may just enter one single image and virtually replicate this one in order to create a whole static sequence.

> NOTE: TO ENCODE FASTER A DCP WITH STATIC IMAGES, SELECT A FOLDER FOR DCDM J2K STORAGE AND SELECT "RESUME FROM DCDM" IN THE ENGINE PANEL. IN THIS CASE YOUR SEQUENCE WILL UNDERGO JUST ONE J2K TRANSFORM SEQUENCE, AND WILL DO A SIMPLE COPY AFTERWARDS.

Moreover, remember to select the proper ***Colorspace*** for your input video clip, else color tones could appear very different from wanted ones when projecting. Generally, BT.709 or sRGB are fine for HD contents.

If you are an expert, you can choose your particular colorspace transformation coefficients, by setting a custom colorspace and filling in the 9 transformation coefficients in the proper control.[36]

Just to make an example, if *C* is your origin color in the source RGB space, then it will be transformed into the destination XYZ colorspace as color *C'*, according to the following relationship

$$\mathbf{C}' = \begin{bmatrix} c'_1 \\ c'_2 \\ c'_3 \end{bmatrix} = \mathbf{TC} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix},$$

[36] The coefficients must not be scaled to take into account the DC headroom specifications: the program will do it automatically. Just provide the direct transformation coefficients from your own origin colorspace to destination XYZ.

and the t_{ij} coefficients are those entered in the GUI control.

The ***Gamma*** field must be left empty; just use with care when you want to force a gamma different from the default colorspace one.

Audio clips

Enter audio clips by right-clicking on the list box and selecting the files you want to include in your project. Similarly, files can be also removed from the lists. Use **+** and **–** keys to raise up/lower down a file in the list.

- If the video clip has an audio content, too, this will be used for DCP audio. Else, a stereophonic silence track is included.
- If an auxiliary multichannel audio content is specified, that will be used.
- If many auxiliary monophonic audio channels are added, they will be used.

Eventually, choose the proper channel layout configuration (up to 16 channels are supported).

A particular care must be made to match, as best as possible, the video frame rates with audio sampling rates. In some particular cases, the program will warn that possible audio glitches will be presented to output. This warning can normally be ignored, but always check your DCP audio before projection (from beginning to end, that is, check that there is no delay sync accumulation).

Subtitle clips

Enter one subtitle file only in SubRip (*.srt*) format, in the list box. The subtitle file must begin with a spot number, i.e. no blank line. The times are associated with the original frame rate of the video clip, thus a change in video frame rate will keep the subtitles synchronized.

Clip formats

At the time of writing, the supported clip (image, video, audio, subtitle) formats are reported in tables 3 to 6. Some of these formats will not give you the full expected precision, so please read also the accompanying notes.

Table 3 - Supported image sequence formats.

Format	Library used for import	Notes
BMP	Libnsbmp[37]	8 bit only

Format	Library used for import	Notes
DPX	OpenDPX[38]	
PNG	LibPNG[39]	8 and 16 bits
SGI	Internal	16 bit OK
TGA	LibTarga[40]	
TIFF	LibTIFF[41]	8, 12, 16 bit (no RGBA)

Table 4 - Supported video formats.

Format	Library used for import	Notes
All video codecs supported by FFmpeg	FFmpeg plugin[42]	8 bit only. Due to patenting and licensing reasons, in order to have extended video support, FFmpeg binaries must be downloaded and installed optionally and separately. You are in charge of satisfying and respecting all royalty-related issues when using licensed codecs in FFmpeg.[43]

Table 5 - Supported audio formats.

Format	Library used for import	Notes
FLAC	libFLAC	
WAV	Internal	8, 16, and 24 bit, PCM format only
All audio codecs supported by FFmpeg	FFmpeg plugin[42]	Due to patenting and licensing reasons, , in order to have extended audio support, FFmpeg binaries can be downloaded and installed optionally and separately. You are in charge of satisfying and respecting all royalty-related issues when using licensed codecs in FFmpeg.[43]

[37] http://www.netsurf-browser.org/projects/libnsbmp/

[38] SMPTE DPX v2 Image Format reader/writer (http://code.google.com/p/dpx/)

[39] http://www.libpng.org/

[40] http://www.cs.wisc.edu/graphics/Gallery/LibTarga/

[41] http://www.libtiff.org/

[42] http://www.ffmpeg.org/

[43] http://www.ffmpeg.org/legal.html, read section "Patent Mini-FAQ"

Table 6 - Supported subtitle formats.

Format	Library used for import	Notes
SRT	Internal[44]	ASCII encoding only, no UNICODE. First character in file must be spot number “1”.

DCDM panel

The DCDM panel (fig. 22) allows the user to select the picture and sound properties of the DCP.

The frame rate of video and sampling rate of audio can be decided in the rates section. There are a bunch of rates available, the most standard and working ones are 24 and 48 FPS, for video, and 48 kHz, for audio.

If you entered a SubRip file in the input section, you can also choose the way to embed subtitles in your DCP. The first one, burn-in, of course, superimposes the text right on the images. The other methods use XML and font files.

The total duration of the DCP can be decided by entering the required number of frame, or the duration in seconds. Specifying a value of 0 frames will use all the available frames in the input source. Simply put, you should always set it to 0, unless you are testing.

Moreover, reels can be created every *x* minutes or having a max storage size or containing a certain number of frames. If you just want one reel, set ***Frames*** to 0. However, when you first add a new clip in the video ***Input*** panel, the number of frames is used to fill this control (i.e., one reel only).

Reels can also be specified by adding a number of points in the ***cue list***: you can enter a timecode or frame number per line, and a new reel will be created at that timecode.

Making multiple reels out of a single input clip is a debated question. One possible answer is that, if you use a FAT 32 formatted device for storing the DCP (a USB key, for instance), your maximum file size is limited to 2 GB.

You can decide the size of your DCP files by adjusting the ***Total*** bitrate value: 250 Mbps is the highest standardized value, so do not exceed to maintain compatibility.

Expert users may also want to change the ***Component*** bitrate: this is the maximum bitrate that each one of the three components can achieve.

In the ***Folders*** section can be entered the paths of the folders that can (optionally) contain the DCDM files. You can drag-and-drop folders over them.

[44] Partial support for SubRip, features like italic text, color change, vertical layout, etc., do not work.

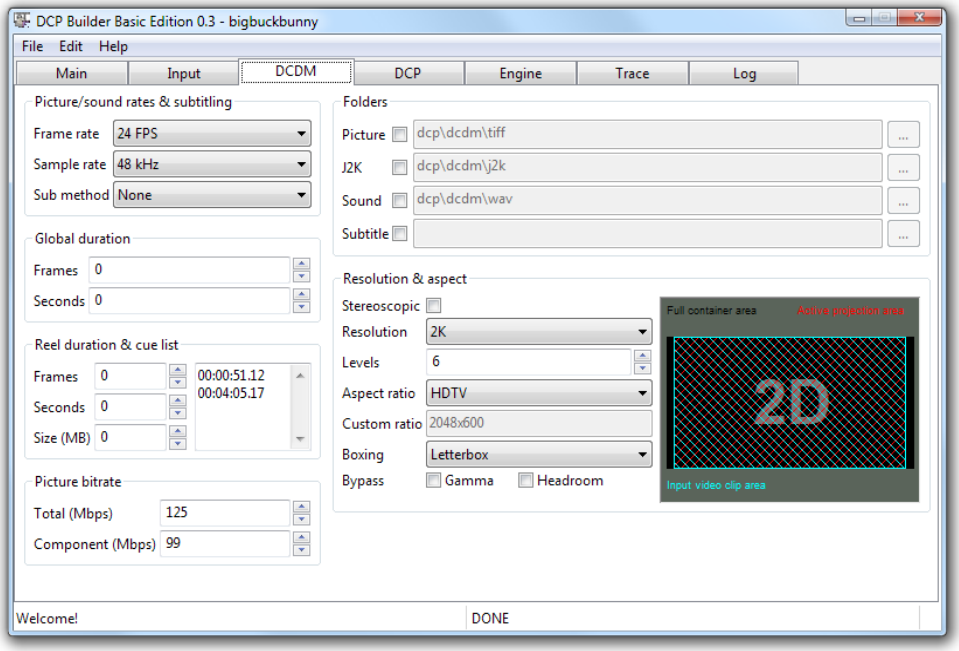


Figure 22 - The DCDM panel.

Stereoscopic DCPs can be created, provided that you enter two input files in the input video section.

You can also decide the resolution and what the best aspect ratio is for your DCP based on the input material, and you get an idea of how the screen is filled. In the aspect ratio screen, the black area represents the full projection matrix (2048x1080 for 2K or 4096x2160 for 4K), the red area is the active projection area (i.e., the actually encoded picture), and the cyan area is the input video clip picture format (fig. 23).

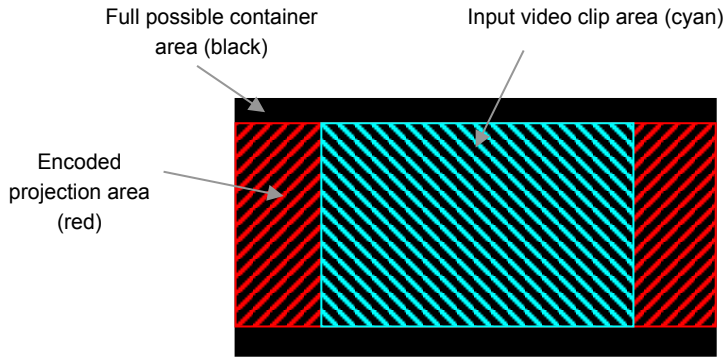


Figure 23 - The aspect ratio layout area.

Eventually, for the more experienced people, there is the option to completely bypass the DC gamma encoding and the digital headroom scaling.[45] Normally, these options are disabled: in order to enable them, you must visit the ***Edit→Preferences*** dialog. Doing so, the DCDM preparation is completely into your hands. Figure 24 reports the difference, in the DC gamma transfer function, when the normalization constant is used or not: the correct transfer function should be the red one.

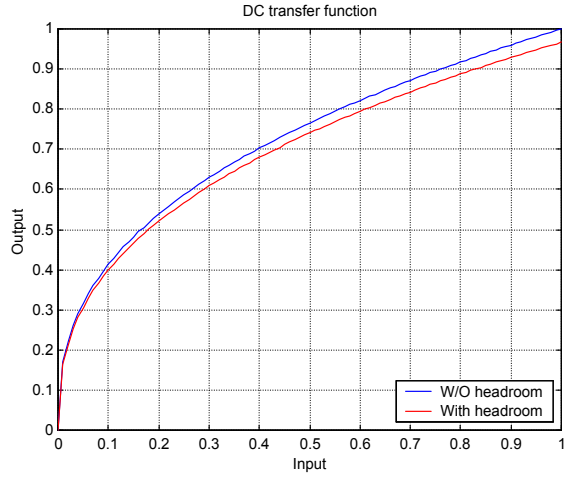


Figure 24 – Normalized and un-normalized DC gamma transfer function.

[45] Usually, if you disable gamma & headroom, you should also set the colorspace to XYZ and the gamma to 1.0 in the Inputs panel.

Common aspect ratios

Figure 25 presents a visual and direct comparison of the most common aspect ratios used in Digital Cinema and HD video, both for the 2K and 4K paradigms.

> WARNING! DON'T USE A CUSTOM ASPECT RATIO UNLESS YOU KNOW WHAT YOU ARE DOING! EVEN IF VALID, CUSTOM ASPECT RATIOS ARE GENERALLY NOT IMPLEMENTED IN MANY PROJECTOR MACRO SETUPS, SO IT SHOULD BE NECESSARY TO ASK THE PROJECTIONIST OR THE INSTALLER TO MODIFY/ADD IT MANUALLY.

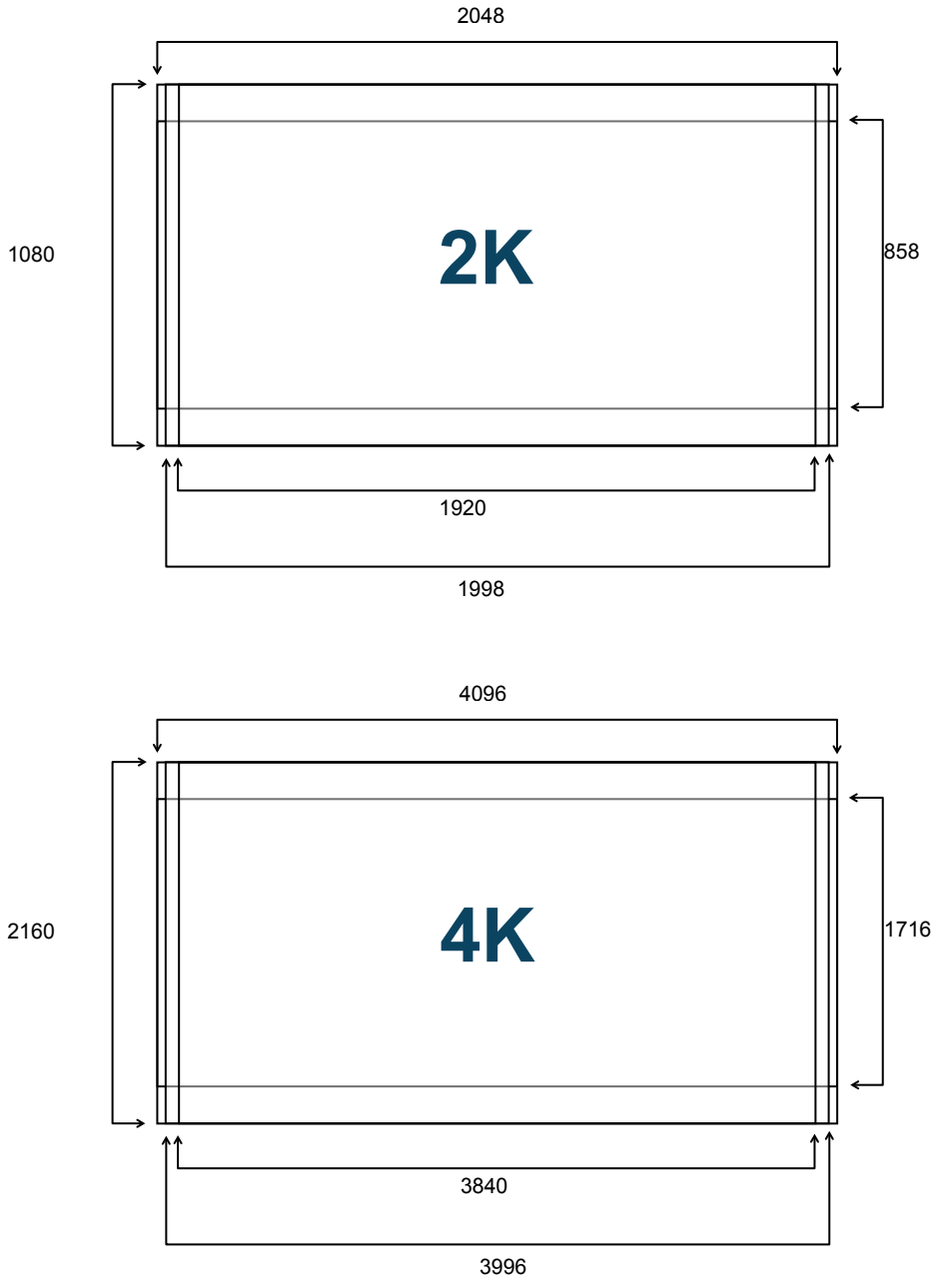


Figure 25 - Comparison of common aspect ratios: 2K Scope (2048x858), 2K Flat (1998x1080), 2K HDTV (1920x1080), 2K Full (2048x1080), 4K Scope (4096x1716), 4K Flat (3996x2160), 4K SHDTV (3840x2160), 4K Full (4096x2160).

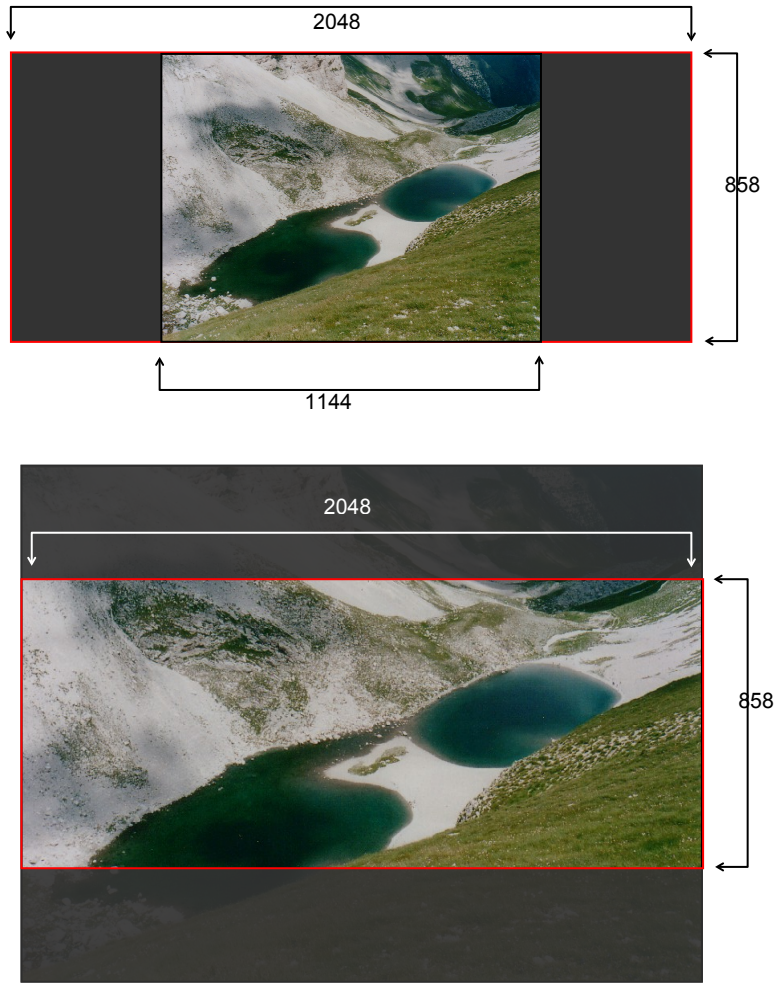


Figure 26 - Rendering of 4:3 SDTV onto Scope in pillarbox (top) and letterbox (bottom) modes. Red border outlines effective projection screen size. N.B.: 4:3 clips (without cropping) should not be rendered in pillarbox scope, but are best done in pillarbox flat.

Figure 27 - Rendering of 16:9 HDTV onto Scope in pillarbox (top) and letterbox (bottom) modes. Red border outlines effective projection screen size. N.B.: 16:9 clips (without cropping) should not be rendered in pillarbox scope, but are best done in pillarbox flat.

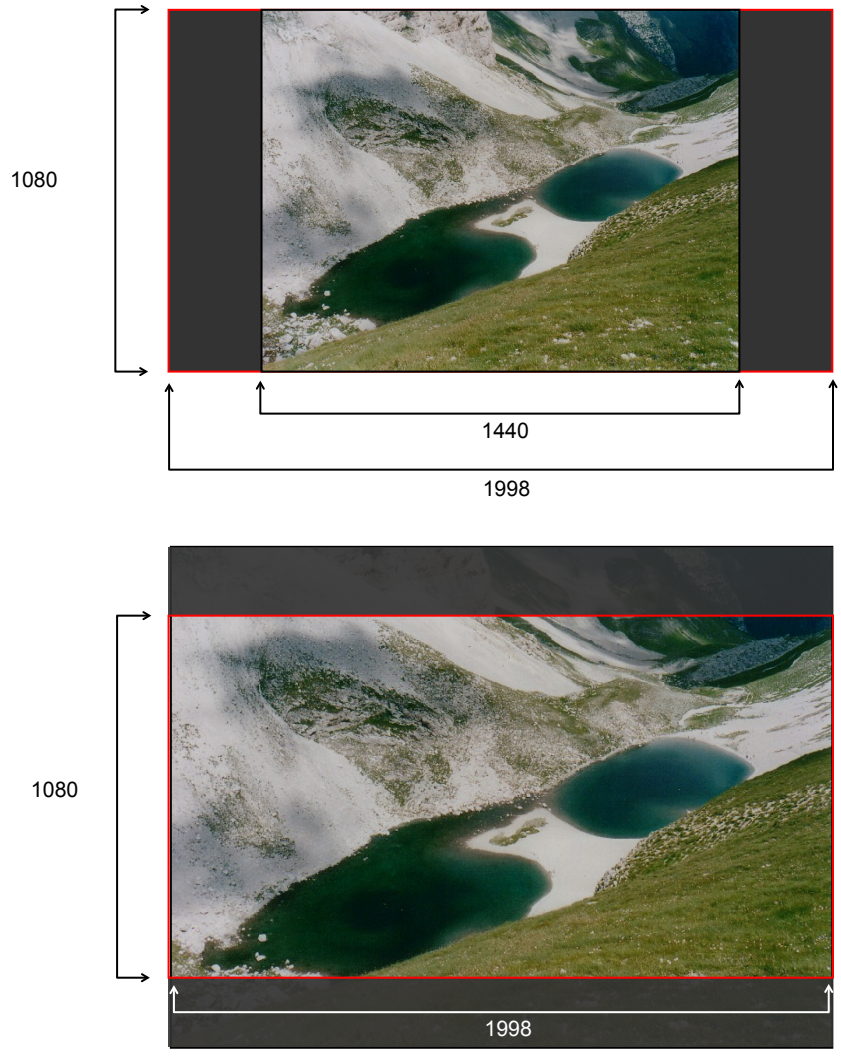


Figure 28 - Rendering of 4:3 SDTV onto Flat in pillarbox (top) and letterbox (bottom) modes. Red border outlines effective projection screen size.

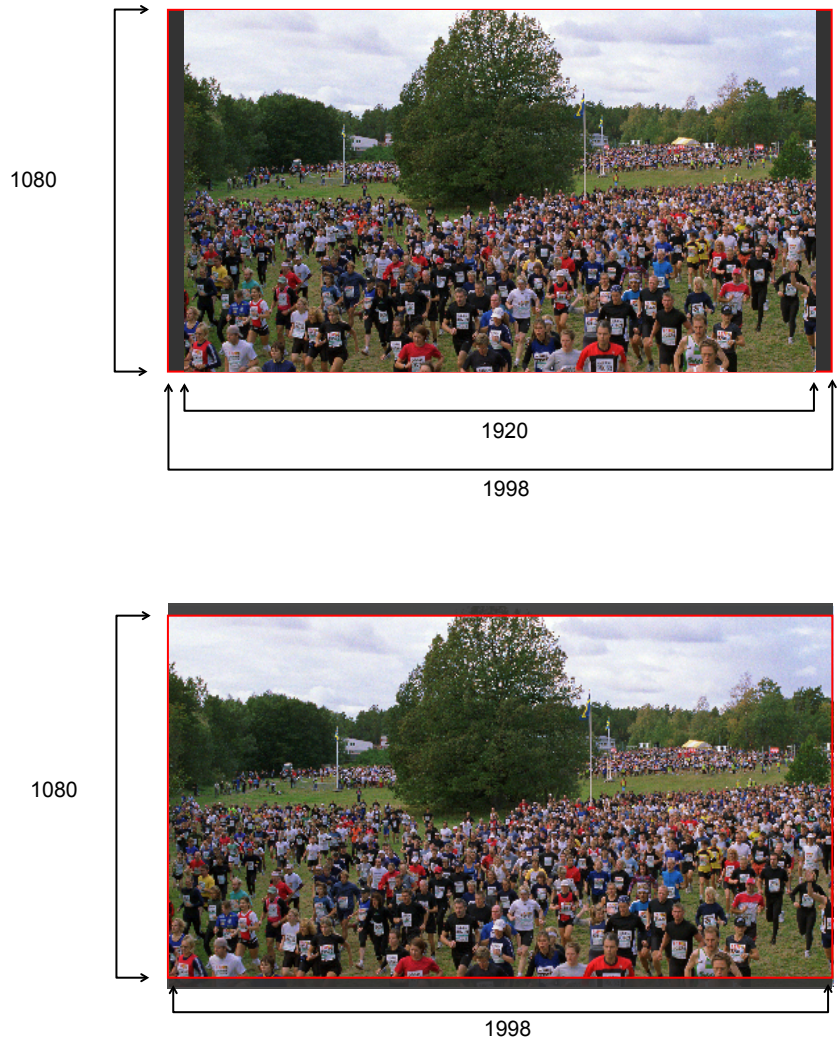


Figure 29 - Rendering of 16:9 HDTV onto Flat in pillarbox (top) and letterbox (bottom) modes. Red border outlines effective projection screen size.

DCP panel

The DCP panel (fig. 30) allows the user to select the format and metadata of the DCP.

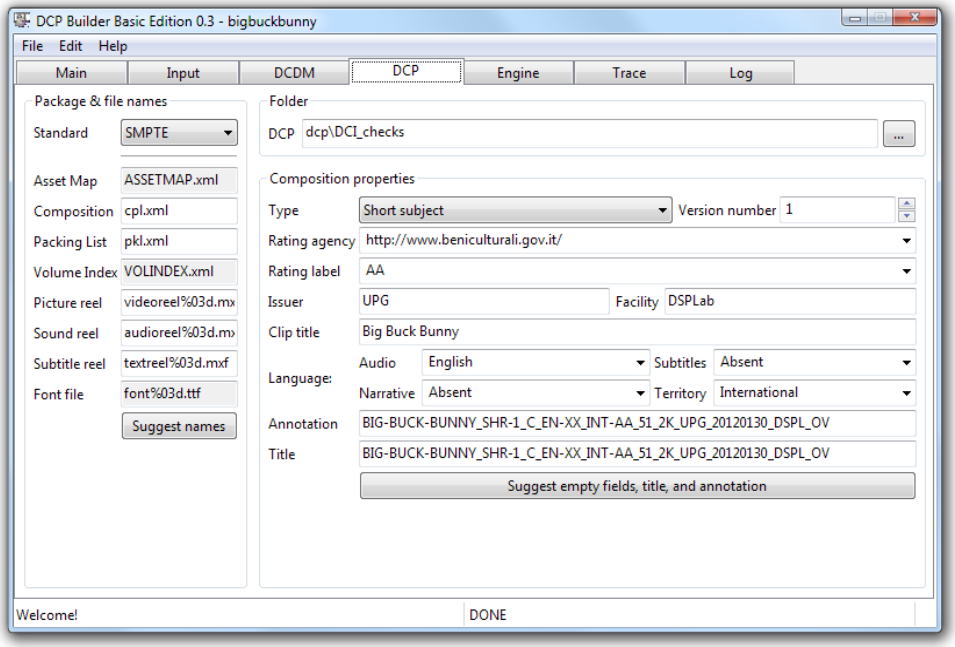


Figure 30 - The DCP panel.

The movie is stored in a folder that you can enter or select in the ***DCP*** text field (accepts drag-and-drop).

In the ***Package & files*** section, the names of MXF reels and XML files are given, whereas in the ***Composition properties*** section, you can fill all the metadata of your DCP. In some cases, a predefined list presents some possible choices, which nonetheless can be overridden with custom values.

There are two large buttons (***Suggest***) that help you to fill automatically some fields.

In ***Package & file names*** section, pressing several times the suggest button will give you different naming schemes for the DCP files. Just accept one of them or, if you are an expert, override them to suit your needs (see manual section "Names of the DCP files [string]" for a more detailed explanation).

In the ***Composition properties*** section, title, and annotation are filled according to the commonly used digital cinema naming convention.[46]

Engine panel

The engine panel (fig. 31) allows the user to select some processing parameters to use during transcoding.

[46] http://www.digitalcinemanamingconvention.com/

For example, ***Resume*** can be used to start encoding from a partially done DCP: if you dump J2Ks, the program reads them[47] and encoding is faster.

The ***Resize*** method is used to adapt original video and audio to the DCP ones:

- Picture: complexity but accuracy grows as you select *copy*, *nearest*, *bilinear*, *lanczos*, *bicubic*, or *bspline*.
- Sound: complexity but accuracy grows as you select *copy*, *nearest*, *linear*, or *sinc*.[48]

The *copy* option should be selected only when you are sure that the origin format is exactly in the proportion and resolution required by your DCP project.

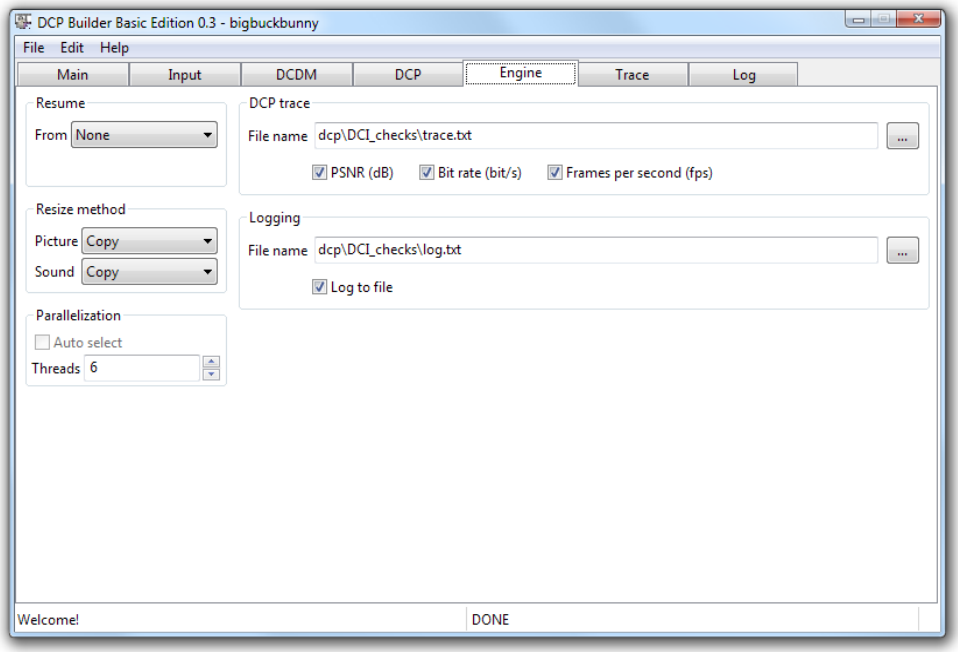


Figure 31 - The engine panel.

Differently from all other controls, which are disabled during DCP creation, the ***Threads*** spin button remains active and allows you to dynamically choose the number of concurrent threads dedicated to compression. For instance, when you need more responsiveness for other tasks done in the encoding PC, you can decrease this value temporarily and increase it back when you are done.

The number of threads is limited as follows:

[47] At the current development stage, the only DCDM resume type happens from the J2K folder.

[48] In this case, the term "sinc" has nothing to do with synchronization.

- there is a maximum of (*virtualCPUs* - 1) possible threads (or 2);
- the maximum limit for threads is 24.[49]

It is also possible to specify the path of a trace file, where information on encoding (PSNR, time, bitrate) is written. This same information is also available on the trace panel, in a nice view.

The trace file is an ASCII text file, with a very simple format, such as

```
...
108   63.8   0   250049472   144232512   66710208   37150080   3.4
120   62.9   0   250023744   148121088   64031424   37605312   3.4
132   61.6   0   250022976   146086464   63366720   40508544   3.4
144   60.2   0   249775872   135135360   72251520   42510528   3.4
156   59.3   0   249834432   134035776   75941568   39974592   3.4
...
```

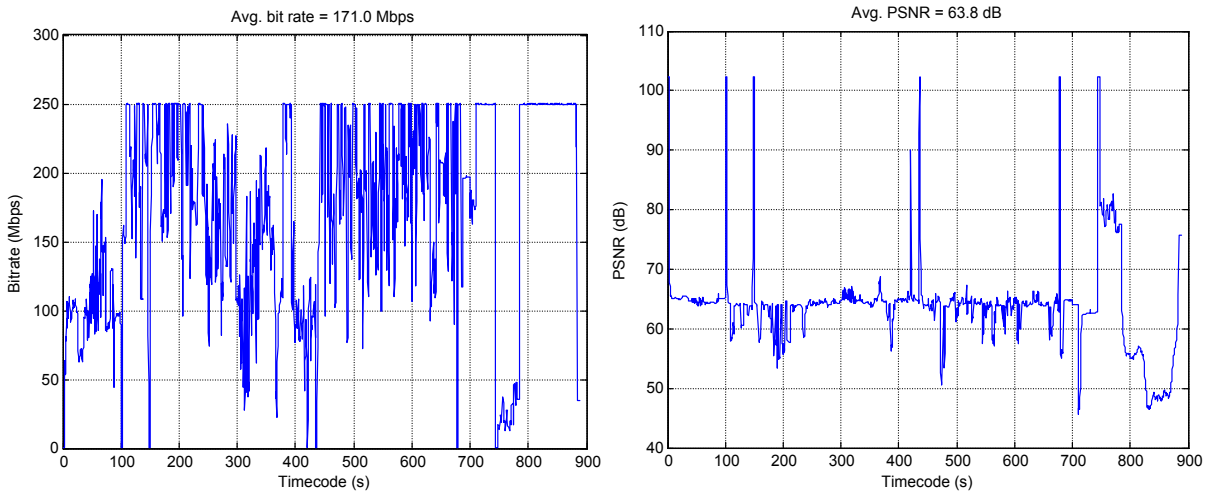
Each line is a tab-separated list of

- frame number,
- PSNR (dB),
- 4K, 2K, and component bit rates (bps),
- frame rate (fps).

This file can be easily post-processed for recovering the encoding history (fig. 32). An *m-script*[50] that can be used for loading and drawing the results shown in the figure is available from the download site.

[49] E.g., dual hex core CPU with hyperthreading. Depends on the software edition.

[50] This script can be used with programs such as Matlab, Octave, or Scilab.



(a) *(b)*

Figure 32 - PSNR (a) and bitrate (b) history as extracted from the trace file. The shown trace is that obtained with the "Sintel" 2K movie frames.[51]

Eventually, you can also specify a log file, where all the logging is also copied into. A sample log is as follows:

```
...
07/06/2011 12:48:02: CTH00: Compression thread starting
07/06/2011 12:48:02: CTH01: Compression thread starting
07/06/2011 12:48:02: CTH02: Compression thread starting
07/06/2011 12:48:02: DTH00: Decompression thread starting
07/06/2011 12:51:27: CTH#00: Created new reel
07/06/2011 12:54:32: CTH#02: Created new reel
07/06/2011 13:07:44: CTH01: Compression thread exiting
07/06/2011 13:07:45: CTH02: Compression thread exiting
07/06/2011 13:07:46: CTH00: Compression thread exiting
07/06/2011 13:07:46: Finalizing DCP (might take a little time)
...
```

Trace panel

The trace panel (fig. 33) hosts a running history view of the DCP encoding process, useful for quality check purposes. The top portion of the panel plots the PSNR, whereas the bottom portion plots the bitrate.

The bitrate plot hosts five different curves:

[51] http://www.sintel.org/

- 4K bitrate, valid only if doing 4K, differently it is set at 0;
- 2K bitrate, that is total for 2K or subtotal for 4K;
- *Y* component bitrate;[52]
- *Cx* component bitrate;
- *Cz* component bitrate.

Left clicking on each panel will pop-up a small menu, allowing to scale, fit, shift, or fix the axes. It is also possible to save a snapshot of every plot as a bmp file.

Figure 33 - The trace panel, showing PSNR (top) and bitrate (bottom) encoding history.

Log panel

The logging panel (fig. 34) allows the user to inspect information emitted during the encoding process.

[52] During JPEG 2000 compression, an Irreversible Color Transform (ICT) is applied to the *XYZ* picture, resulting in a different colorspace, *YCxCz*.

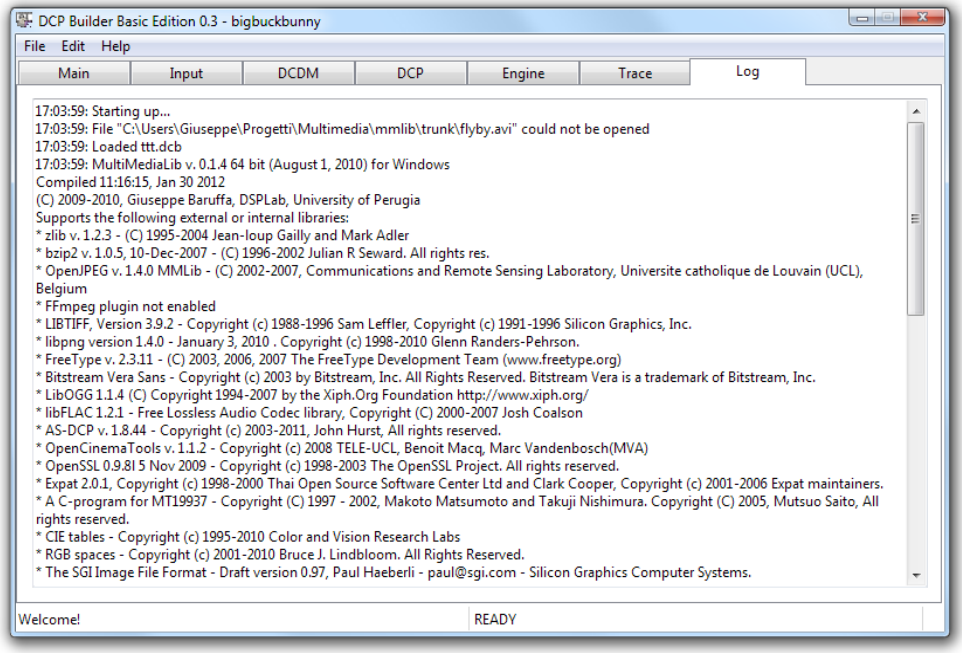

DCP Builder Basic Edition 0.3 - bigbuckbunny
File Edit Help
Main Input DCDM DCP Engine Trace Log
17:03:59: Starting up...
17:03:59: File "C:\Users\Giuseppe\Progetti\Multimedia\mmlib\trunk\flyby.avi" could not be opened
17:03:59: Loaded ttt.dcb
17:03:59: MultiMediaLib v. 0.1.4 64 bit (August 1, 2010) for Windows
Compiled 11:16:15, Jan 30 2012
(C) 2009-2010, Giuseppe Baruffa, DSPLab, University of Perugia
Supports the following external or internal libraries:
* zlib v. 1.2.3 - (C) 1995-2004 Jean-loup Gailly and Mark Adler
* bzip2 v. 1.0.5, 10-Dec-2007 - (C) 1996-2002 Julian R Seward. All rights res.
* OpenJPEG v. 1.4.0 MMLib - (C) 2002-2007, Communications and Remote Sensing Laboratory, Universite catholique de Louvain (UCL), Belgium
* FFmpeg plugin not enabled
* LIBTIFF, Version 3.9.2 - Copyright (c) 1988-1996 Sam Leffler, Copyright (c) 1991-1996 Silicon Graphics, Inc.
* libpng version 1.4.0 - January 3, 2010 . Copyright (c) 1998-2010 Glenn Randers-Pehrson.
* FreeType v. 2.3.11 - (C) 2003, 2006, 2007 The FreeType Development Team (www.freetype.org)
* Bitstream Vera Sans - Copyright (c) 2003 by Bitstream, Inc. All Rights Reserved. Bitstream Vera is a trademark of Bitstream, Inc.
* LibOGG 1.1.4 (C) Copyright 1994-2007 by the Xiph.Org Foundation http://www.xiph.org/
* libFLAC 1.2.1 - Free Lossless Audio Codec library, Copyright (C) 2000-2007 Josh Coalson
* AS-DCP v. 1.8.44 - Copyright (c) 2003-2011, John Hurst, All rights reserved.
* OpenCinemaTools v. 1.1.2 - Copyright (c) 2008 TELE-UCL, Benoit Macq, Marc Vandenbosch(MVA)
* OpenSSL 0.9.8l 5 Nov 2009 - Copyright (c) 1998-2003 The OpenSSL Project. All rights reserved.
* Expat 2.0.1, Copyright (c) 1998-2000 Thai Open Source Software Center Ltd and Clark Cooper, Copyright (c) 2001-2006 Expat maintainers.
* A C-program for MT19937 - Copyright (C) 1997 - 2002, Makoto Matsumoto and Takuji Nishimura. Copyright (C) 2005, Mutsuo Saito, All rights reserved.
* CIE tables - Copyright (c) 1995-2010 Color and Vision Research Labs
* RGB spaces - Copyright (c) 2001-2010 Bruce J. Lindbloom. All Rights Reserved.
* The SGI Image File Format - Draft version 0.97, Paul Haeberli - paul@sgi.com - Silicon Graphics Computer Systems.
Welcome!
READY

Figure 34 - The log panel.

At startup, the logger shows all the loaded and compiled libraries.

Every event is time-stamped.

If errors are popped up during normal operation, please check this log window for inspecting details.

Menu bar

File menu

Commands in the ***File*** menu (fig. 35) are used to clean, initialize, close, save, or open an existing set of parameters, called *project* (with extension *.dcb*). And, of course, to shut down the application.

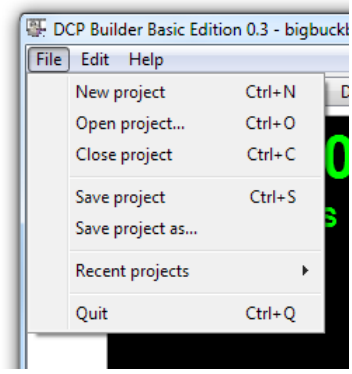


Figure 35 - The File menu.

New project menu

You can either start from a brand new set of parameters, with all fields clean, or follow the indications of a simple wizard.

Just answer to the simple questions (and follow the hints) that you will be asked, and you'll finish with functional project settings (fig. 36).

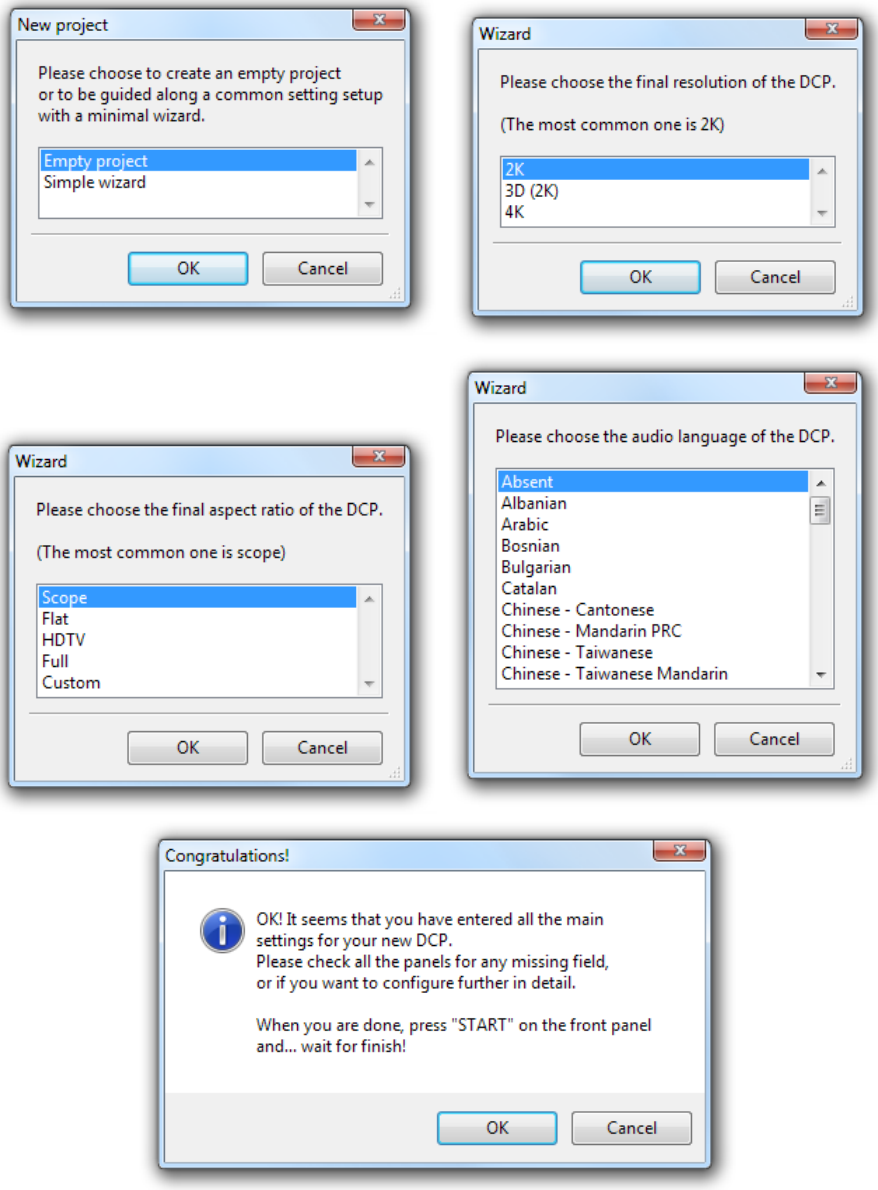


Figure 36 - Some steps of the simple wizard procedure.

Edit menu

Commands in the ***Edit*** menu (fig. 37) are used to set program preferences and to enable/disable the processing quality preview image.

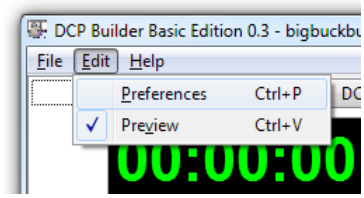


Figure 37 - The Edit menu.

Preferences menu

Some default program preferences can be setup via the ***Edit→Preferences*** menu.

A dialog will allow changing these parameters:

- ***Aspect*** (fig. 38-a)
  - ***LCD background color***: the color to be used in the LCD background (default is black).
  - ***LCD foreground color***: the color to be used in the LCD foreground (default is green).
  - ***Layout***: change from single window to multiple window layout. Application restart is required.
- ***Input*** (fig. 38-b)
  - BE ***Enable FFmpeg support***: tick if you want to enable support for video input files. This requires that you have installed or compiled a recent version of FFmpeg.[53] For further details, read the section "Video import in Basic edition".
  - BE ***FFmpeg folder***: this is the folder where the *ffmpeg* and *ffprobe* executables can be found. If the programs are in your search path, leave this field empty.

> NOTE: IF YOU SPECIFY THE EXECUTABLES FOLDER, ALWAYS APPEND THE TRAILING SLASH OR BACKSLASH IN YOUR PATH, E.G.,
>
> C:\PROGRAM FILES\FFMPEG\BIN\
> /OPT/FFMPEG/BIN/
> /APPLICATIONS/FFMPEG/BIN/

- ***DCP*** (fig. 38-c)
  - ***Issuer***: default issuer name for your DCPs.
  - ***Facility***: default rating label for your DCPs.
  - ***Rating label***: default rating label for your DCPs.

[53] The program has been tested with FFmpeg git-14d94a1.

- ***Rating agency***: default rating agency for your DCPs.
- ***Prompt on overwrite***: it will always prompt before overwriting files in the DCP folder.
- ***Create folder***: if there are output folders that do not yet exist, they will be created (however, hierarchical folder creation is not available).
- ***Custom filenames***: you may override the standard recommendations and select custom names for volume index and asset map files. Clearly, doing so will render the DCP unplayable. Application restart is required.

- ***DCDM*** (fig. 38-d)
  - ***White point chromaticity adaptation***: the color transformation matrix can be weighted in order to adapt the origin colorspace white point to the SMPTE DCDM white point. This is done by means of a Bradford-type cone response. I do not understand if that is really necessary from a "colorist" point-of-view, so instead of applying it by default, the choice is left to you. I expect feedback on this.
  - ***Enable gamma/headroom bypass***: if you want to create the DCDM entirely on your own responsibility, you can disable DCI gamma encoding and digital headroom in the DCDM panel. For doing so, you must enable this option here.
- ***Preview*** (fig. 38-e)
  - ***Frame decode interval***: increase or decrease the interval between frames that will also be fully decoded in order to obtain a preview. A good trade-off is 1 or 2 previews (interval set to 24 or 12) for each second of clip.

> NOTE: THIS IS ALSO THE FRAME PERIOD AT WHICH BIT RATE AND PSNR ARE UPDATED IN THE LCD PANEL AND IN THE TRACE FILE.

  - ***XYZ->RGB transform & gamma***: activate real colors on your preview panel, slightly decelerating the encoding speed.
- ***Other*** (fig. 38-f)
  - ***Send mail when done***: if the encoding job takes a lot of time, and you are far from your PC, you can get informed by a summary email when the task is accomplished.
  - ***SMTP data***: in order for the email to be sent, you have to provide a valid SMTP server address and port, in the form *server_name:port_number* (usually, port number is 25, but you have to specify it anyway). You can also customize the *From* and *To* fields of the email. The received email will contain some details on the encoding task (title, reels, size, encoding duration, etc.) and, if enabled, the log file of the encoding as an attachment (fig. 39).

NOTE: THE SMTP SERVER MUST NOT REQUIRE ANY SORT OF AUTHENTICATION, OTHERWISE NO EMAIL WILL BE SENT.

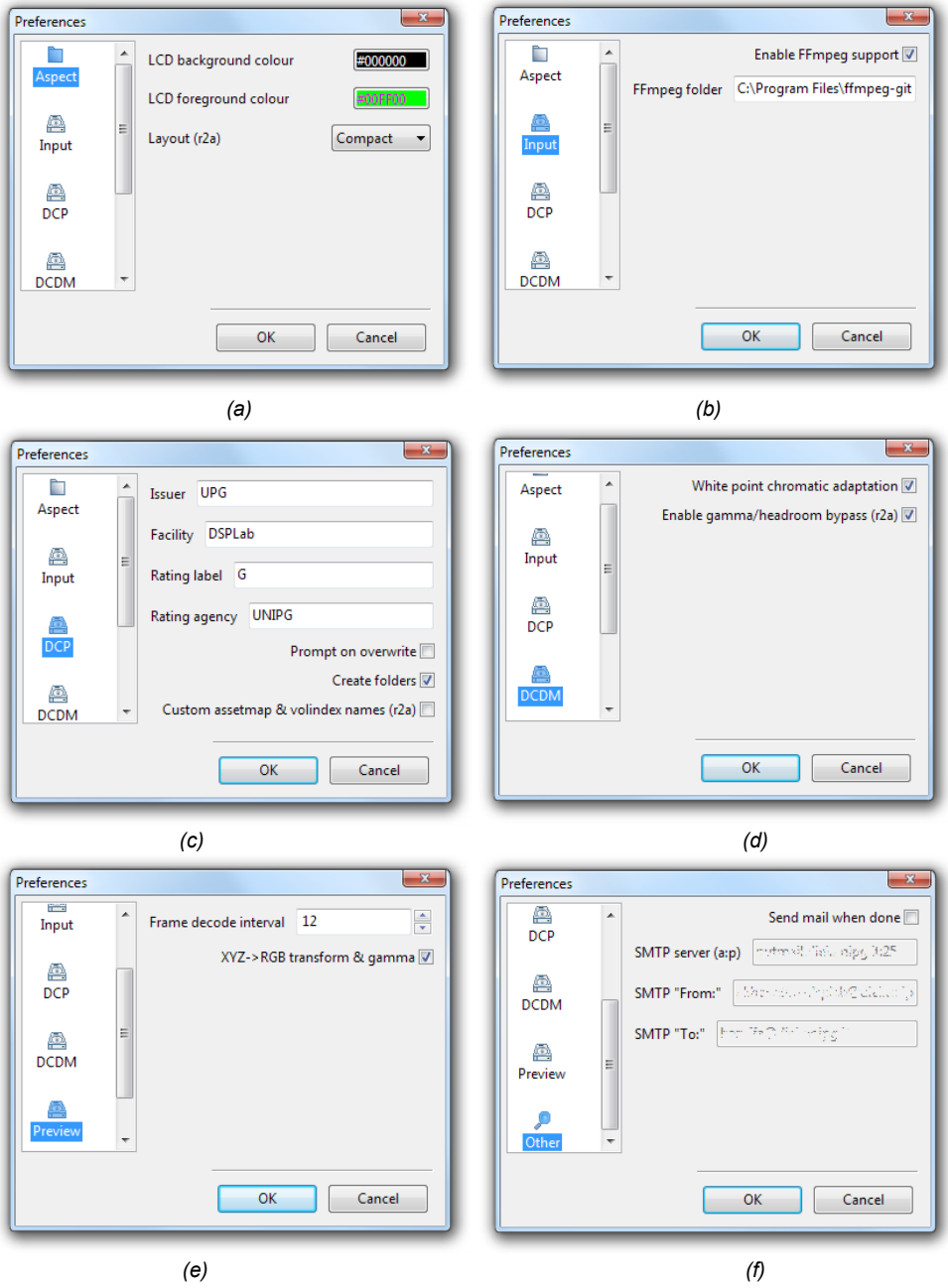


Figure 38 - The Preferences dialog.

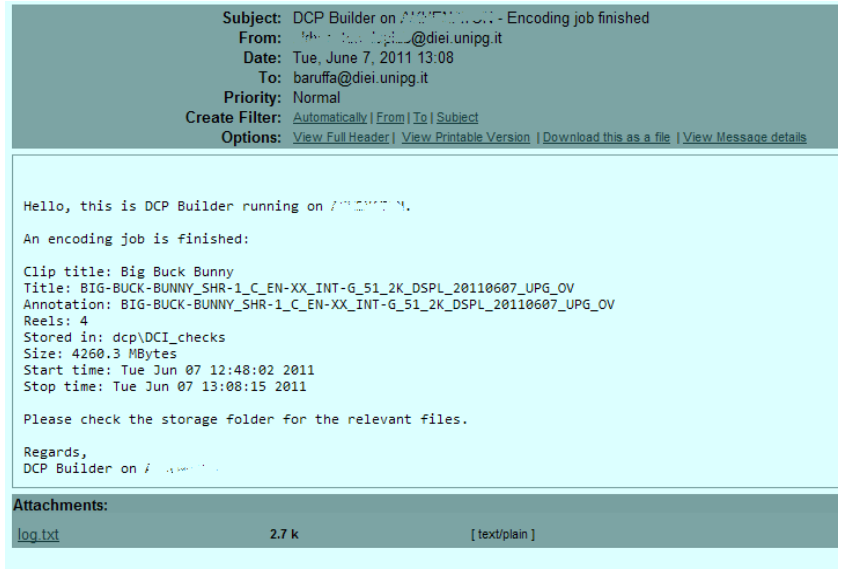

Subject: DCP Builder on [redacted] - Encoding job finished
From: [redacted]@diei.unipg.it
Date: Tue, June 7, 2011 13:08
To: baruffa@diei.unipg.it
Priority: Normal
Create Filter: Automatically | From | To | Subject
Options: View Full Header | View Printable Version | Download this as a file | View Message details
Hello, this is DCP Builder running on [redacted].
An encoding job is finished:
Clip title: Big Buck Bunny
Title: BIG-BUCK-BUNNY_SHR-1_C_EN-XX_INT-G_51_2K_DSPL_20110607_UPG_OV
Annotation: BIG-BUCK-BUNNY_SHR-1_C_EN-XX_INT-G_51_2K_DSPL_20110607_UPG_OV
Reels: 4
Stored in: dcp\DCI_checks
Size: 4260.3 MBytes
Start time: Tue Jun 07 12:48:02 2011
Stop time: Tue Jun 07 13:08:15 2011
Please check the storage folder for the relevant files.
Regards,
DCP Builder on [redacted]
Attachments:
log.txt 2.7 k [text/plain]

Figure 39 - The email received at the end of an encoding job.

View menu

This menu appears only when expanded mode layout is selected (fig. 40). It can be used to show or hide all the accessory panels (project settings, preview, log, trace) and to take them all on top of the windows stack at once.

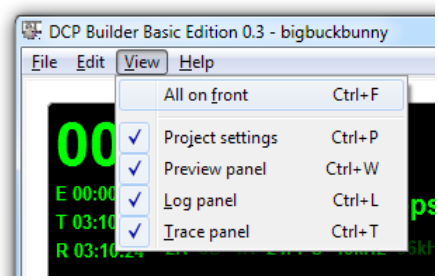


Figure 40 - The View menu.

In this case, the aspect of the main panel is that of fig. 9, and the preview lives in a separate window (fig. 42).

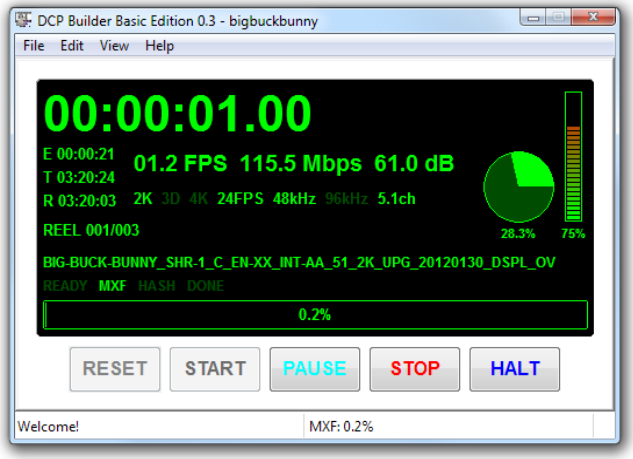


Figure 41 - The main panel in expanded layout.

Figure 42 - The preview panel.

In the preview panel, a popup menu allows choosing several things:

- ***Fullscreen***: go full screen with the preview. If you have a second monitor, you can use it.

- ***Zoom***: ***Fit*** lets the preview always touch from inside the preview panel windows borders, or you can select a variable zoom (***25%***, ***50%***, ***75%***, and ***100%***) and drag with the mouse to explore several parts of the picture.
- ***Stereo***: choose between previewing the ***Left*** or the ***Right*** view.

There is a small one-line summary of encoding quality on the top-left side of the preview: in this case, the reported FPS are averaged on the last 100 frames, thus you can have a running complexity estimation.

Help menu

Commands in the ***Help*** menu (fig. 43) are used to show generic information on the program version and author(s), as well as the credits to all the software libraries adopted. A PDF manual (this document) can also be opened.

Moreover, it is also possible to enter the registration code that removes the watermark imprinted on all the images in the rendered DCP. For more info on registration, see “Registration” section.

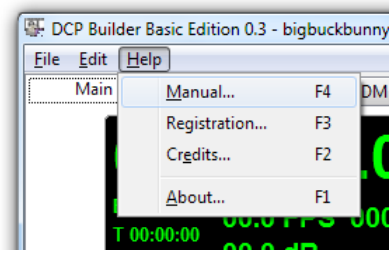


Figure 43 - The Help menu.

Other windows are those relevant to ***About*** (fig. 44) and ***Credits*** (fig. 45).

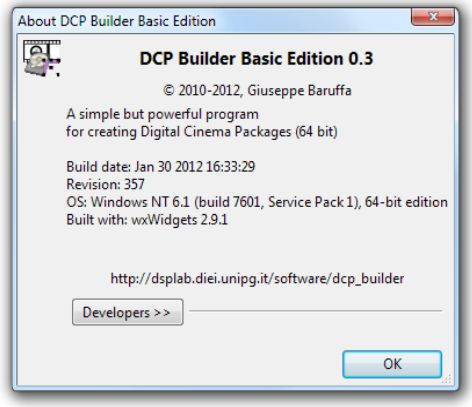

About DCP Builder Basic Edition
DCP Builder Basic Edition 0.3
© 2010-2012, Giuseppe Baruffa
A simple but powerful program
for creating Digital Cinema Packages (64 bit)
Build date: Jan 30 2012 16:33:29
Revision: 357
OS: Windows NT 6.1 (build 7601, Service Pack 1), 64-bit edition
Built with: wxWidgets 2.9.1
http://dsplab.diei.unipg.it/software/dcp_builder
Developers >>
OK

Figure 44 - The About window.

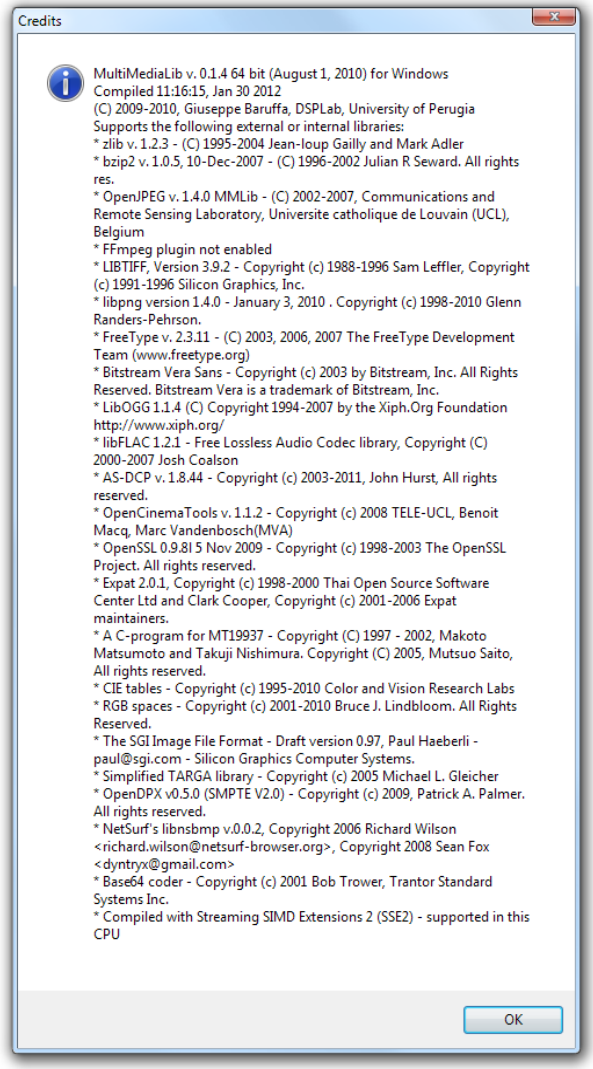


Figure 45 - The Credits window.

Status bar

The status bar shows a summary of the menu being selected and the current processing status (fig. 46).

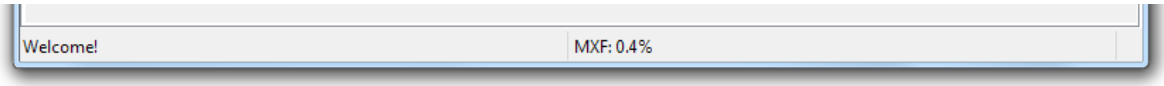


Figure 46 - The status bar.

How-To's

How to build a DCP

Well, it's time for you to create a DCP.

You have several options, such as being guided by the easy wizard

- Select ***File→New project→Simple wizard***

or choosing between a number of predefined setups

- Select ***File→New project→2K 24 fps Interop*** (or similar names)

Differently, if you prefer a completely manual configuration, follow these steps:

- Start the program, and select ***File→New project→Empty project***.
- Add a video clip in the ***Input*** panel video section.[54]
- Select the proper origin ***Colorspace***
    - BT.709 is acceptable for HD video footage shot with professional camcorders; sRGB may be good for synthetic contents such as cartoons or computer-generated scenery (e.g. Blender).
    - Write down a ***Gamma*** value only if you want to use one that is different from the selected colorspace one, otherwise let the space blank.
- If the video clip has no audio, and you want no sound, the created DCP will have a stereophonic silence.
  Otherwise, if the video clip has also audio, this will be used.
  If you want to add a different multichannel audio, then add the file in the input panel audio sections.
  If you have a group of monophonic files, one per channel, add them.
- Remember to select the proper audio channel configuration (indeed, this is automatically inferred by the added files, but this selector is required for improved compatibility).

[54] If the clip consists of a series of images in a folder, you have to select the proper frame rate before adding the clip.

- If you want subtitles, you can either burn them in or add them as XML files in CineCanvas™ format. Add a SubRip (*.srt*) subtitle file in the input field and choose the proper method of embedding.
- Select size and rates in the ***DCDM*** panel.
    - 2K, scope, letterbox, 24 fps, and 48 kHz is the most common format. Choose the best aspect ratio and boxing for your clip.
    - If you have subtitles, choose the burn-in method or the *Interop XML+font* method (the other methods are not fully tested and could not work).
    - Do not enable picture, sound, j2k, or subtitle folders, if you don't need a DCDM or DCDM*.
    - Set global duration to 0 frames for encoding the whole clip.
    - Set reel size to 0 frames, to have a single reel.

> HINT: IF YOU STORE THE DCP ON A FAT32 FORMATTED DISK, MAKE REELS AND SET THEIR SIZE TO A VALUE LOWER THAN 2 GB.

    - Set bitrate to a value lower than the standardized maximum 250 Mbps. Values in the range from 100 to 180 Mbps generally produce a good visual result, with a lower bitrate. Values higher than 250 Mbps might not work on certain servers (it's not standardized).
- Choose the characteristics in the ***DCP*** panel.
    - *Interop* is the old standard (but some servers still support only this[55]), whereas *SMPTE* is the newest one. Use ***Suggest names*** to let the system fill automatically the other fields.
    - Select the folder where the DCP will be stored (please check before that enough room is available in that disk).
    - Choose the type of the DCP, write down the rating agency and label, and the issuer. Then, select the languages. Eventually, press ***Suggest title & annotation*** to fill the last two fields with common conventions. Since the naming convention is a bit complicated sometimes, you may need to check it and correct accordingly.[56]
      Anyhow, a DCP with an off-standard naming convention will always work on every server.

[55] For example, the Dolby DSS100 with old firmware does not support SMPTE but only Interop.

[56] See http://www.digitalcinemanamingconvention.com/ for a more complete and thorough guidance to fill correctly this field.

- Set the encoding parallelization in the ***Engine*** panel.
    - You can begin by setting a low value, e.g., 2. You can modify this value as the encoding goes. Do not choose a number higher than the number of CPUs/hyperthreads on your machine.
    - Set ***Resume*** to None.
    - If the picture and sound do not need resampling, leave *Nearest* as method in both selectors. You can use *Copy* if the frame rate, sample rate, and image size of inputs are exactly the same of the DCP.
- Press ***Start*** button in the main panel to begin encoding and wait for finish.
    - The **MXF** label is lit during the main encoding phase; this means that the *.mxf* files containing the audio and video reels are being produced. You can have an estimate of how long the encoding will last (if you have an SMTP server available, you can receive an email at the end).
    - The **HASH** label is lit nearly at the end of the whole process. The possibly multi-gigabyte reel files are being scanned and their checksum is written in a set of *.xml* files. During this phase, the timing labels are not working (but the slider is), just be patient and wait for finish (for full features, this step takes even 15-20 minutes, depending on your HD and CPU speed).
    - Do not stop or abort anything up to the blinking **DONE** label.
- If you want to make a short test, press ***Stop*** when you want to finalize the DCP at the timecode shown in the LCD panel.[57]

How to encode video clips

Starting from version 0.3, it is no more possible to import in a straightforward way any video format.

Up to version 0.2, this was possible on the fly, by using the FFmpeg/Libav libraries.[58] However, this support has been discontinued, for patenting reasons. In fact, even if FFmpeg or Libav were compiled in LGPL compatibility (it means that a closed source program, such as DCP Builder, can use open source programs as external, dynamically loaded library), this does not relief one's responsibility to satisfy all the licenses for the involved patented technologies. Most notably, they are H.264, mp3, and MPEG-2.

This support can now be available via separate install of FFmpeg, which must be downloaded by you.

---

[57] Please note that some servers might refuse to play DCPs shorter than 1 second.

[58] http://www.ffmpeg.org/, http://www.libav.org/

Video import in Freeware Edition

In this case, if you need to convert a video clip file into a DCP, I suggest you to extract all the frames from the video clip as a sequence of PNG images and as a single multichannel WAV file. This is what you have to do:

- Download a compiled version of FFmpeg or Libav for your operating system. Generally, these programs can be found via Google, or starting from the project home pages.
- Once you have the *ffmpeg* executable working, open a terminal and go to folder where you want to make the conversion step.
- Create the folders *extrpng* and *extrwav*, or similar names.
- Execute the command in order to extract all the frames to the PNG format

```
#> ffmpeg -i my_clip.avi extrpng/my_clip_%06d.png
```

- Execute the command in order to extract all the audio channels to a WAV file

```
#> ffmpeg -i my_clip.avi -acodec pcm_s24le extrwav/my_clip.wav
```

- You now have a sequence of images (the frames of your input video clip) in the *extrpng* folder, and a single multichannel wave file in the *extrwav* folder. You can use these files as input to DCP Builder.

Video import in Basic edition

In this case, if you need to convert a video clip file into a DCP, you must explicitly enable FFmpeg support into DCP Builder. There are two ways: binaries download, or source download and compilation. I have tested the program with *ffmpeg-git-14d94a1* version. Any newer version should be ok, for older ones nothing is guaranteed.

Binary downloads

Download a compiled version of FFmpeg or Libav for your operating system. Generally, these programs can be found via Google, or starting from the project home pages. Note that you need both the *ffmpeg* and *ffprobe* programs, preferably compiled statically in order to avoid shared libraries mismatch.

Microsoft Windows

A good place to look at, is Zeranoe's website (google it).

GNU Linux

You can use the official or unofficial repositories of your distribution. For instance, under Ubuntu, you can do the following:

```
#> sudo apt-get install ffmpeg ffprobe
```

Apple Mac OS X

Precompiled binaries are hard to find, but you can have a chance by looking at the ffmpegX project. Browse inside the ffmpegX app and look for ffmpeg executables.

Differently, a semi-automatic method to get the program installed is to use macports or fink.

Source download and compilation

Install manually the program executables into a folder of your choice. You might also do a system-wide install.

Microsoft Windows

- Download the FFmpeg release tarball (e.g., *ffmpeg-git-14d94a1.tar.bz2*) from its download page.
- Open the compressed archive and extract its contents to a folder (e.g., *c:\Program Files\ffmpeg-git-14d94a1*). You can use 7zip.[59]
- Download and install *Visual C++ 2008 Express Edition with SP1* from Microsoft page.[60]
- Download and run the most recent MinGW installer (e.g., *mingw-get-inst-20111118.exe*) from the MinGW page.[61]
- During the setup phase, select to install *C compiler* and *MSYS Basic System*. Wait for all packages being downloaded and installed automatically, some minutes are needed.
- Download generic Win32 Yasm executable (e.g., *yasm-1.2.0-win32.exe*) from Yasm download page.[62] Rename it to *yasm.exe* and copy it inside the *bin* folder of your MinGW installation directory (e.g., *c:\MinGW\bin*).
- Launch *MinGW Shell* from your Windows menu.

[59] http://www.7zip.org/

[60] http://www.microsoft.com/visualstudio/en-us/products/2008-editions/express

[61] http://www.mingw.org/

[62] http://yasm.tortall.net/

- In the just opened MINGW32 console, type *lib* and verify that you obtain an output containing "Microsoft (R) Library Manager". Differently, ...
- Change to the folder where you extracted the FFmpeg archive,

```
#> cd c:/Program\ Files/ffmpeg-git-14d94a1
```

- Create directories *build* and *inst*, and change folder to *build*

```
#> mkdir build inst & cd build
```

- Now you need to configure FFmpeg. Since you are compiling by yourself, and experimenting for personal use, you can enable almost anything. When you have tested this, you have to recompile according to patent restrictions issued in your country. Thus, issue the configure command and let it run for some minutes:

```
#>  ../configure  --prefix=../inst  --enable-nonfree  --enable-gpl  --
   disable-devices --disable-doc --disable-ffplay --disable-ffserver --
   enable-w32threads  --enable-memalign-hack  --disable-debug  --enable-
   static --disable-shared
```

- At this point, you can issue the compilation commands (this can take a long time, so please be patient and wait)

```
#> make clean & make
...
[lines of compilation results]
...
#> make install
```

- Now, if no compilation errors appeared, check inside the *inst* folder of FFmpeg directory: in the *bin* subfolder there will be all the required programs (*ffmpeg* and *ffprobe*)

```
#> ls ../inst/bin
ffmpeg  ffprobe
```

- You can now indicate this folder in program preferences.

GNU Linux

- Download the FFmpeg release tarball (e.g., *ffmpeg-git-14d94a1.tar.bz2*) from its download page.

- Open the compressed archive and extract its contents to a folder (e.g., */tmp/ffmpeg-git-14d94a1*)

```
#> cd /tmp
#> tar zxvf ffmpeg-git-14d94a1.tar.bz2
```

- Install the required compilation tools. For instance, under Ubuntu,

```
#> sudo apt-get update & sudo apt-get install build-essential yasm
   zlib1g-dev
```

- Change to the folder where you extracted the FFmpeg archive

```
#> cd /tmp/ffmpeg-git-14d94a1
```

- Create directories *build* and */opt/ffmpeg* and change folder to build

```
#> mkdir build /opt/ffmpeg & cd build
```

- Now you need to configure FFmpeg. Since you are compiling by yourself, and experimenting for personal use, you can enable almost anything. When you have tested this, you have to recompile according to patent restrictions issued in your country. Thus, issue the configure command and let it run for some minutes:

```
#> ../configure --prefix=/opt/ffmpeg --enable-nonfree --enable-gpl --
   disable-devices --disable-doc --disable-ffplay --disable-ffserver --
   enable-w32threads --enable-memalign-hack --disable-debug --enable-
   static --disable-shared
```

- At this point, you can issue the compilation commands (this can take a long time, so please be patient and wait)

```
#> make clean & make
...
[lines of compilation results]
...
#> sudo make install
```

- Now, if no compilation errors appeared, check inside the */opt/ffmpeg* folder: in the *bin* subfolder there will be all the required programs (*ffmpeg* and *ffprobe*)

```
#> ls /opt/ffmpeg/bin
```

```
ffmpeg  ffprobe
```

- You can now indicate this folder in program preferences.

Apple Mac OS X

- Download the FFmpeg release tarball (e.g., *ffmpeg-git-14d94a1.tar.bz2*) from its download page.
- Using the terminal, open the compressed archive and extract its contents to a folder (e.g., */tmp/ffmpeg-git-14d94a1*)

```
#> cd /tmp
#> tar zxvf ffmpeg-git-14d94a1.tar.bz2
```

- Install the required compilation tools, namely *Xcode*. Insert your Mac OS X operating system DVD in the tray, and look into the *Optional installs* folder. Then, double click on *Xcode.mpkg* and follow the installation instructions.
- Change to the folder where you extracted the FFmpeg archive

```
#> cd /tmp/ffmpeg-git-14d94a1
```

- Create directories *build* and */opt/ffmpeg* and change folder to build

```
#> mkdir build /opt/ffmpeg & cd build
```

- Now you need to configure FFmpeg. Since you are compiling by yourself, and experimenting for personal use, you can enable almost anything. When you have tested this, you have to recompile according to patent restrictions issued in your country. Thus, issue the configure command and let it run for some minutes:

```
#>  ../configure  --prefix=/opt/ffmpeg  --enable-nonfree  --enable-gpl  --
   disable-devices --disable-doc --disable-ffplay --disable-ffserver --
   enable-w32threads  --enable-memalign-hack  --disable-debug  --enable-
   static --disable-shared --disable-yasm
```

- At this point, you can issue the compilation commands (this can take a long time, so please be patient and wait)

```
#> make clean & make
...
```

```
[lines of compilation results]
...
#> sudo make install
```

- Now, if no compilation errors appeared, check inside the */opt/ffmpeg* folder: in the *bin* subfolder there will be all the required programs (*ffmpeg* and *ffprobe*)

```
#> ls /opt/ffmpeg/bin
ffmpeg  ffprobe
```

- You can now indicate this folder in program preferences.

How to play a DCP without a server or projector

When your DCP is ready, you may want to test before ingesting it into the DC server. There are some possibilities in this case, mainly using a software player.

For purposes of syntactic correctness of the DCP, I suggest you to use Fraunhofer's *EasyDCP player*,[63] which has a limited playtime of 15 s, and no color transform, but is great for checking if the XML files are fine.

If you need to preview your full DCP, then you can use digitAll's *dcpPlayer*,[64] which is quite fast but has a limited time license and watermarks the image.

Another option is to use 3dtv.at's *Stereoscopic player*,[65] which limits the play-out to 5 minutes but has no time expiration. Anyway, this is very cheap software which you can afford easily.

[63] http://www.dcinema.fraunhofer.de/en/veranstaltungen/ibc/player.jsp

[64] http://digitall.net.au/products/dcp-player.html

[65] http://www.3dtv.at/Index_en.aspx

Command line user interface

In addition to the GUI, DCP Builder has also a command line interface (CLI) that allows producing a DCP with exactly the same quality, without need of windows or physical screens (fig. 47-a, -b, and -c).

The CLI reads the projects saved by the GUI and processes the files exactly as usual, by providing the file as a parameter:

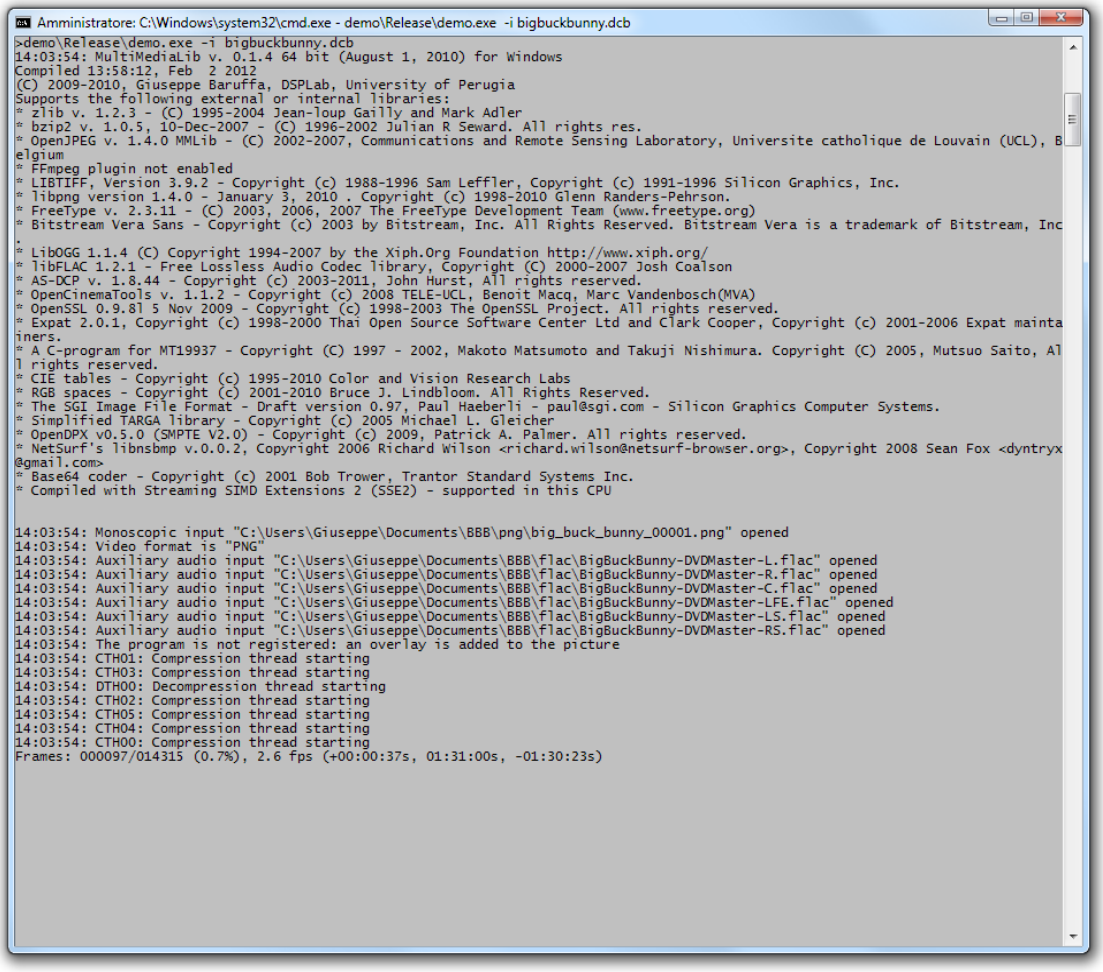
```
#> dcpbuilder [-i project_file_name] [-R registration_code] [-h] [-smtps
   smtp_server_name:port]    [-smtpf    smtp_from]    [-smtpt    smtp_to]
   [-sf search_folder]
```

The parameters are:

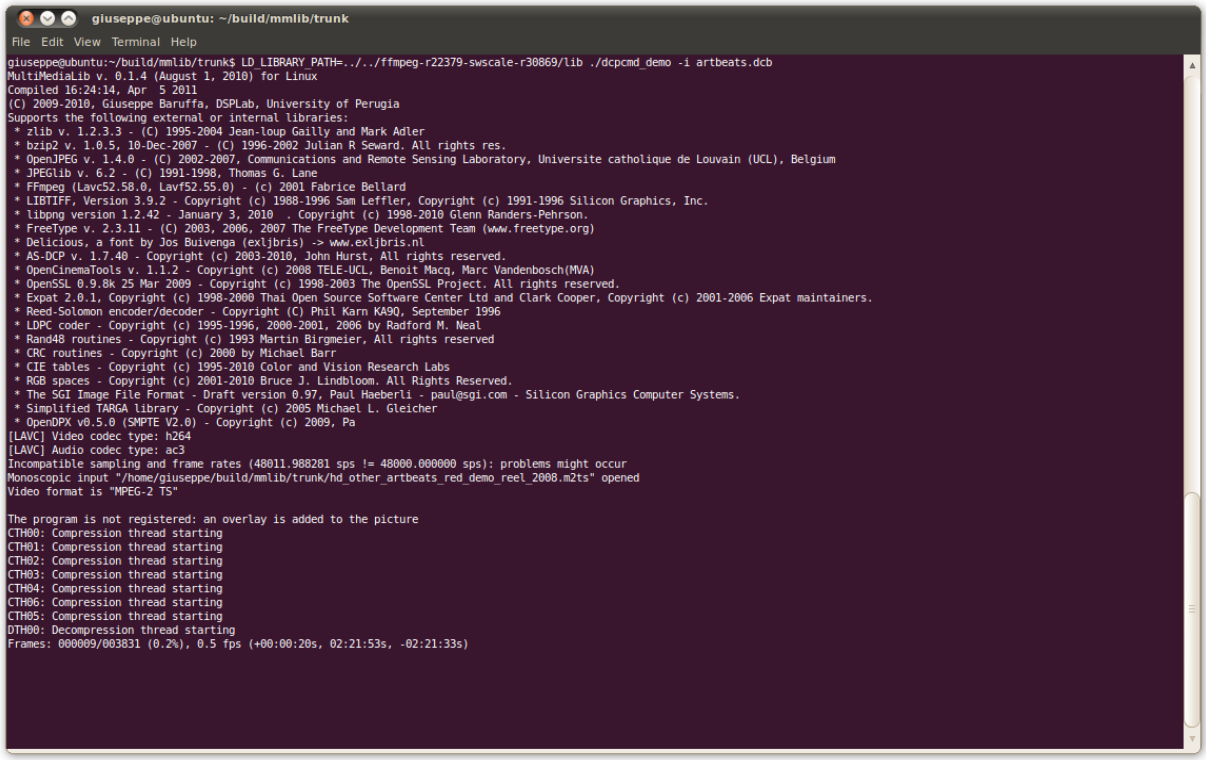
- *-i project_file_name*, accepts the input project file. By default, it is *dcp_builder_sample.dcb*.
- *-R registration_code*, enter the received serial to remove the watermark.
- *-smtps smtp_server_name:port*, enter the name and port of a SMTP server used to send emails.
- *-smtpf smtp_from*, enter the "From" address of emails.
- *-smtpt smtp_to*, enter the "To" address of emails.
- *-sf search_folder*, folder where the *ffmpeg* executables are looked for. Remember to terminate the path with a trailing slash or backslash.
- *-h*, prints help and usage.

During encoding, pressing "*h*" will show a mini-menu of commands available to the user:

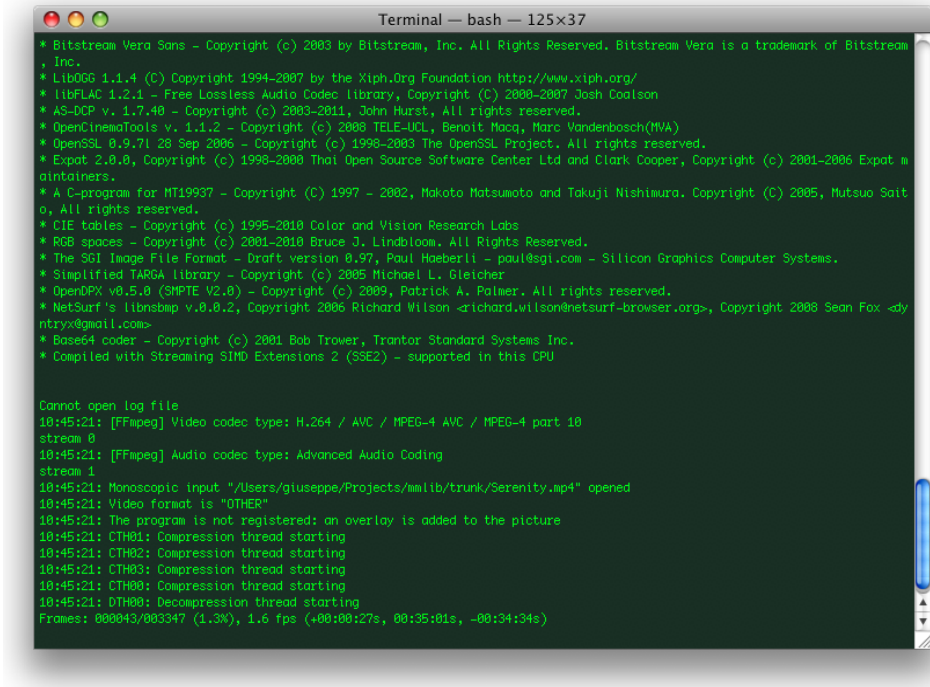
- *h* - prints help
- *i* - prints information
- *s* - prints information
- *+* - add a thread
- *-* - remove a thread
- *q* - quits the encoding at the current point (equivalent to pressing ***STOP*** in the GUI)


Amministratore: C:\Windows\system32\cmd.exe - demo\Release\demo.exe -i bigbuckbunny.dcb
>demo\Release\demo.exe -i bigbuckbunny.dcb
14:03:54: MultiMediaLib v. 0.1.4 64 bit (August 1, 2010) for Windows
Compiled 13:58:12, Feb 2 2012
(C) 2009-2010, Giuseppe Baruffa, DSPLab, University of Perugia
Supports the following external or internal libraries:
* zlib v. 1.2.3 - (C) 1995-2004 Jean-loup Gailly and Mark Adler
* bzip2 v. 1.0.5, 10-Dec-2007 - (C) 1996-2002 Julian R Seward. All rights res.
* OpenJPEG v. 1.4.0 MMLib - (C) 2002-2007, Communications and Remote Sensing Laboratory, Universite catholique de Louvain (UCL), B
elgium
* FFmpeg plugin not enabled
* LIBTIFF, Version 3.9.2 - Copyright (c) 1988-1996 Sam Leffler, Copyright (c) 1991-1996 Silicon Graphics, Inc.
* libpng version 1.4.0 - January 3, 2010 . Copyright (c) 1998-2010 Glenn Randers-Pehrson.
* FreeType v. 2.3.11 - (C) 2003, 2006, 2007 The FreeType Development Team (www.freetype.org)
* Bitstream Vera Sans - Copyright (c) 2003 by Bitstream, Inc. All Rights Reserved. Bitstream Vera is a trademark of Bitstream, Inc
.
* LibOGG 1.1.4 (C) Copyright 1994-2007 by the Xiph.Org Foundation http://www.xiph.org/
* libFLAC 1.2.1 - Free Lossless Audio Codec library, Copyright (C) 2000-2007 Josh Coalson
* AS-DCP v. 1.8.44 - Copyright (c) 2003-2011, John Hurst, All rights reserved.
* OpenCinemaTools v. 1.1.2 - Copyright (c) 2008 TELE-UCL, Benoit Macq, Marc Vandenbosch(MVA)
* OpenSSL 0.9.8l 5 Nov 2009 - Copyright (c) 1998-2003 The OpenSSL Project. All rights reserved.
* Expat 2.0.1, Copyright (c) 1998-2000 Thai Open Source Software Center Ltd and Clark Cooper, Copyright (c) 2001-2006 Expat mainta
iners.
* A C-program for MT19937 - Copyright (C) 1997 - 2002, Makoto Matsumoto and Takuji Nishimura. Copyright (C) 2005, Mutsuo Saito, Al
l rights reserved.
* CIE tables - Copyright (c) 1995-2010 Color and Vision Research Labs
* RGB spaces - Copyright (c) 2001-2010 Bruce J. Lindbloom. All Rights Reserved.
* The SGI Image File Format - Draft version 0.97, Paul Haeberli - paul@sgi.com - Silicon Graphics Computer Systems.
* Simplified TARGA library - Copyright (c) 2005 Michael L. Gleicher
* OpenDPX v0.5.0 (SMPTE V2.0) - Copyright (c) 2009, Patrick A. Palmer. All rights reserved.
* NetSurf's libnsbmp v.0.0.2, Copyright 2006 Richard Wilson <richard.wilson@netsurf-browser.org>, Copyright 2008 Sean Fox <dyntryx
@gmail.com>
* Base64 coder - Copyright (c) 2001 Bob Trower, Trantor Standard Systems Inc.
* Compiled with Streaming SIMD Extensions 2 (SSE2) - supported in this CPU
14:03:54: Monoscopic input "C:\Users\Giuseppe\Documents\BBB\png\big_buck_bunny_00001.png" opened
14:03:54: Video format is "PNG"
14:03:54: Auxiliary audio input "C:\Users\Giuseppe\Documents\BBB\flac\BigBuckBunny-DVDMaster-L.flac" opened
14:03:54: Auxiliary audio input "C:\Users\Giuseppe\Documents\BBB\flac\BigBuckBunny-DVDMaster-R.flac" opened
14:03:54: Auxiliary audio input "C:\Users\Giuseppe\Documents\BBB\flac\BigBuckBunny-DVDMaster-C.flac" opened
14:03:54: Auxiliary audio input "C:\Users\Giuseppe\Documents\BBB\flac\BigBuckBunny-DVDMaster-LFE.flac" opened
14:03:54: Auxiliary audio input "C:\Users\Giuseppe\Documents\BBB\flac\BigBuckBunny-DVDMaster-LS.flac" opened
14:03:54: Auxiliary audio input "C:\Users\Giuseppe\Documents\BBB\flac\BigBuckBunny-DVDMaster-RS.flac" opened
14:03:54: The program is not registered: an overlay is added to the picture
14:03:54: CTH01: Compression thread starting
14:03:54: CTH03: Compression thread starting
14:03:54: DTH00: Decompression thread starting
14:03:54: CTH02: Compression thread starting
14:03:54: CTH05: Compression thread starting
14:03:54: CTH04: Compression thread starting
14:03:54: CTH00: Compression thread starting
Frames: 000097/014315 (0.7%), 2.6 fps (+00:00:37s, 01:31:00s, -01:30:23s)

(a)


giuseppe@ubuntu: ~/build/mmlib/trunk
File Edit View Terminal Help
giuseppe@ubuntu:~/build/mmlib/trunk$ LD_LIBRARY_PATH=../../ffmpeg-r22379-swscale-r30869/lib ./dcpcmd_demo -i artbeats.dcb
MultiMediaLib v. 0.1.4 (August 1, 2010) for Linux
Compiled 16:24:14, Apr 5 2011
(C) 2009-2010, Giuseppe Baruffa, DSPLab, University of Perugia
Supports the following external or internal libraries:
* zlib v. 1.2.3.3 - (C) 1995-2004 Jean-loup Gailly and Mark Adler
* bzip2 v. 1.0.5, 10-Dec-2007 - (C) 1996-2002 Julian R Seward. All rights res.
* OpenJPEG v. 1.4.0 - (C) 2002-2007, Communications and Remote Sensing Laboratory, Universite catholique de Louvain (UCL), Belgium
* JPEGlib v. 6.2 - (C) 1991-1998, Thomas G. Lane
* FFmpeg (Lavc52.58.0, Lavf52.55.0) - (c) 2001 Fabrice Bellard
* LIBTIFF, Version 3.9.2 - Copyright (c) 1988-1996 Sam Leffler, Copyright (c) 1991-1996 Silicon Graphics, Inc.
* libpng version 1.2.42 - January 3, 2010 . Copyright (c) 1998-2010 Glenn Randers-Pehrson.
* FreeType v. 2.3.11 - (C) 2003, 2006, 2007 The FreeType Development Team (www.freetype.org)
* Delicious, a font by Jos Buivenga (exljbris) -> www.exljbris.nl
* AS-DCP v. 1.7.40 - Copyright (c) 2003-2010, John Hurst, All rights reserved.
* OpenCinemaTools v. 1.1.2 - Copyright (c) 2008 TELE-UCL, Benoit Macq, Marc Vandenbosch(MVA)
* OpenSSL 0.9.8k 25 Mar 2009 - Copyright (c) 1998-2003 The OpenSSL Project. All rights reserved.
* Expat 2.0.1, Copyright (c) 1998-2000 Thai Open Source Software Center Ltd and Clark Cooper, Copyright (c) 2001-2006 Expat maintainers.
* Reed-Solomon encoder/decoder - Copyright (C) Phil Karn KA9Q, September 1996
* LDPC coder - Copyright (c) 1995-1996, 2000-2001, 2006 by Radford M. Neal
* Rand48 routines - Copyright (c) 1993 Martin Birgmeier, All rights reserved
* CRC routines - Copyright (c) 2000 by Michael Barr
* CIE tables - Copyright (c) 1995-2010 Color and Vision Research Labs
* RGB spaces - Copyright (c) 2001-2010 Bruce J. Lindbloom. All Rights Reserved.
* The SGI Image File Format - Draft version 0.97, Paul Haeberli - paul@sgi.com - Silicon Graphics Computer Systems.
* Simplified TARGA library - Copyright (c) 2005 Michael L. Gleicher
* OpenDPX v0.5.0 (SMPTE V2.0) - Copyright (c) 2009, Pa
[LAVC] Video codec type: h264
[LAVC] Audio codec type: ac3
Incompatible sampling and frame rates (48011.988281 sps != 48000.000000 sps): problems might occur
Monoscopic input "/home/giuseppe/build/mmlib/trunk/hd_other_artbeats_red_demo_reel_2008.m2ts" opened
Video format is "MPEG-2 TS"
The program is not registered: an overlay is added to the picture
CTH00: Compression thread starting
CTH01: Compression thread starting
CTH02: Compression thread starting
CTH03: Compression thread starting
CTH04: Compression thread starting
CTH06: Compression thread starting
CTH05: Compression thread starting
DTH00: Decompression thread starting
Frames: 000009/003831 (0.2%), 0.5 fps (+00:00:20s, 02:21:53s, -02:21:33s)

(b)



(c)

Figure 47 - The Windows (a), Linux (b), and Mac OS X (c) command line interface.

Technical aspects overview

Technical specifications of DCPs

The DCP root folder (in the storage medium) contains a number of files, some of them used to store the image and audio contents, and some other used to organize and manage the whole playlist.

Picture MXF files

Picture contents are stored in one or more reels corresponding to one or more MXF files. Each reel contains pictures as MPEG-2 or JPEG 2000 essence, depending on the adopted codec. MPEG-2 is no longer compliant with the DCI specification. JPEG 2000 is the only accepted compression format.

- Supported frame rates are 24, 25, 30, 48, 50, and 60 frames per second. Maximum frame size is 2048x1080 for 2K DC (24, 25, 30, 48, 50, and 60 frames per second), and 4096x2160 for 4K DC (24, 25, and 30 frames per second).[66] Common formats are:
    - Flat (1998x1080 or 3996x2160), ~1.85:1 aspect ratio
    - Scope (2048x858 or 4096x1716), ~2.39:1 aspect ratio
- 12 bits per pixel precision (36 bits total)
- XYZ colorspace
- Maximum bit rate is 250 Mbps (1.3 MBytes per frame at 24 fps)

Sound MXF files

Sound contents are stored in reels, too, corresponding to picture reels in number and duration. In case of multilingual features, separate reels are required to convey different languages. Each file contains linear PCM essence.

- Sampling rate is 48,000 or 96,000 samples per second
- Sample precision of 24 bits
- Linear mapping (no companding)

[66] The frame rates of 25, 30, 50, and 60 fps have been added later and may not (yet) be supported on all playback equipment.

- Up to 16 independent channels.

XML files

Asset Map file

List of all files included in the DCP, in XML format.

Composition Playlist file

The presentation order is saved in XML format in this file; each picture and sound reel is identified by its UUID.

Packing List file

All files in the composition are hashed and their hash is stored here, in XML format. This file is generally used during ingestion in DC server to verify if data have been corrupted or tampered in some way.

Volume Index file

A single DCP may be stored in more than one medium (e.g., multiple hard disks). This file is used to identify the volume order in the series.

Stereoscopic DCPs

The very same format shown before is also used to store stereoscopic (3D) contents. In this case, 48, 50, or 60 frames per second are used, with frames alternating for left eye and right eye pictures.[67] Since the maximum bit rate is always 250 Mbit/s, this results in a net 125 Mbit/s for single frame, but the visual quality decrease is generally unnoticeable.

The XYZ colorspace

The XYZ colorspace is used since it can act as a reference colorspace for all devices be them either cameras or projectors. This fact guarantees that the reproduced color tones will be identical to those of the original clip. Fig. 48 shows the CIE XYZ colorspace and how the RGB gamut is enclosed in it.

[67] The speeds of 50 and 60 fps may not yet be supported on all playback equipment.

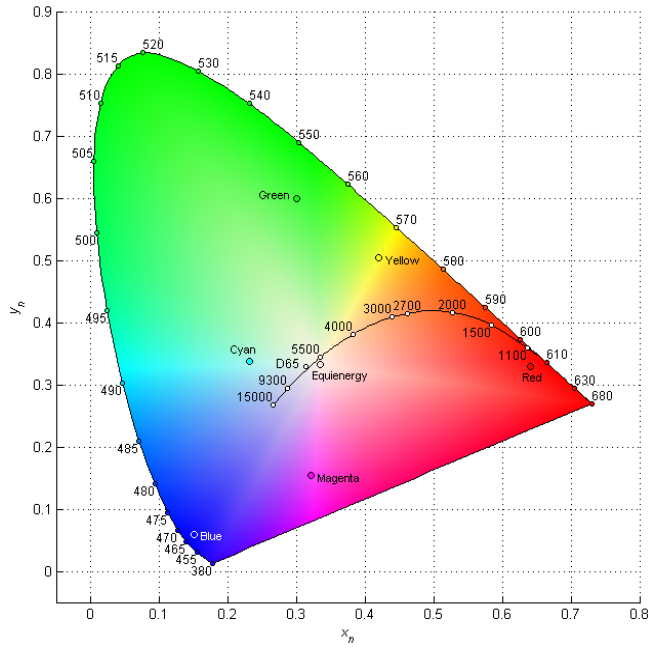


Figure 48 - The CIE XYZ chromaticity diagram with some notable points.

As an example, figures 49 and 50 show how one frame of the Big Buck Bunny movie is rendered in the original sRGB colorspace, and how it appears after transformation into XYZ colorspace. the shallower colors do not pose any problems, since during rendering at the projector, an opposite color transform is done.

Each component sample of the XYZ frame is assumed to be represented with 12 bits: data alignment to byte boundaries is done by moving these 12 bits towards the top of a two bytes 16 bits word.

Figure 49 - The frame no. 2441 of Big Buck Bunny in sRGB colorspace.

Figure 50 - The frame transformed in XYZ colorspace and rendered in RGB colorspace.

How JPEG 2000 works

This section wants to be a concise but technically correct review of how JPEG 2000 standard works to compress digital cinema images.

Lossy coding

A first part of the standard is involved into achieving a first compression target by doing lossy operations, i.e., a sequence of steps that cannot be exactly inverted during decompression. If done judiciously, however, this has no visual impact on the final image quality.

Pixel value normalization

Every sample *x* in every component is normalized in a way to be represented in the [-0.5, +0.5] amplitude interval. With 12 bits, this means that a normalized sample x_{norm} is given by

$$x_{norm} = \frac{x}{4096} - 0.5$$

.

Irreversible Color Transform (ICT)

Another color transform is applied, in order to better decorrelate color components and achieve improved compression efficiency. This color transform is practically equivalent to the well known *RGB→YCbCr* color transform, only that it is applied to an XYZ source; the destination colorspace is generally denoted as *YCxCz*. A representation in the sRGB colorspace of the transformed picture is shown in figure 51.

Figure 51 - The frame transformed in YCxCz colorspace and rendered in RGB colorspace.

Discrete Wavelet Transform (DWT)

This transform is used to decorrelate the image samples, but using a technique which presents some interesting side effects. The trick is that of splitting the image into its horizontal, vertical, and diagonal spatial frequency components. This is achieved via a mathematical object called a Wavelet. However, for most of us, that will simply coincide with a linear filtering operation: low-pass (blurring or averaging) and high-pass (sharpening or differencing), followed by size reduction by a factor of two. For instance, let's consider a 4x4 image component neighborhood with pixel amplitudes

$$\begin{bmatrix} 0.2 & 0.4 \\ -0.4 & 0.0 \end{bmatrix}$$

and its row-wise average

$$\begin{bmatrix} 0.3 \\ -0.2 \end{bmatrix}.$$

Note how the following neighborhood

$$\begin{bmatrix} -0.1 \\ -0.2 \end{bmatrix},$$

can be used, by adding and subtracting the above values, to obtain back the original values. Thus, our simple "horizontal" transform has given us

$$\begin{bmatrix} 0.2 & 0.4 \\ -0.4 & 0.0 \end{bmatrix} \Rightarrow \left[\begin{array}{c|c} 0.3 & -0.1 \\ -0.2 & -0.2 \end{array}\right] .$$

The same operation can be applied this time column-wise, to obtain

$$\left[\begin{array}{c|c} 0.3 & -0.1 \\ -0.2 & -0.2 \end{array}\right] \Rightarrow \left[\begin{array}{c|c} 0.05 & -0.15 \\ \hline 0.25 & 0.05 \end{array}\right] .$$

Thus, the value 0.05 represents the low-pass filtering in the diagonal direction, -0.15 the high-pass in the horizontal, 0.25 the high-pass in the vertical, and 0.05 the high-pass in the diagonal: these components are called the LL, HL, LH, and HH subbands, respectively.

In the general case, averaging and differencing are applied to the whole frame, and the filtering kernel is a 9-7 Cohen-Daubechies-Feauveau wavelet pair.[68] Figure 52 shows decomposition of the BBB frame.

Figure 52 - The frame after one DWT level to achieve two embedded resolutions: the subband limits are highlighted. Clockwise, from top left, there is the LL subband, the LH subband, the HL subband, and the HH subband.

[68] Cohen-Daubechies-Feauveau wavelet. (n.d.). In *Wikipedia*. Retrieved February 3, 2012, from http://en.wikipedia.org/wiki/Cohen-Daubechies-Feauveau_wavelet .

The transform operation can be recursively applied to the LL image subband, in order to achieve a so-called dyadic decomposition. Figure 53 shows a 4-level, 5-resolutions decomposition of the BBB frame.

Figure 53 - The frame after DWT into 5 resolutions: the subband limits are highlighted.

Dead-zone quantization

The subband samples are quantized in sign-magnitude representation, allowing for a dead-zone around 0. The number of quantization levels is responsible for a slower or faster execution of the following processing steps.

Sample data partitioning

The original or transformed coefficients are grouped into a series of successively nested partitions:

- *Tiles*: images in JPEG 2000 standard may be split in several tiles before compression. DCI standard requires that only one tile is present, the whole image only.
- *Components*: the tile is considered as a group of three components. DCI states them to always represent *Y*, *Cx*, and *Cz* colors (ICT is mandatory).
- *Resolutions*: a resolution is a group of three high-pass subbands (HL, LH, and HH), at higher levels, or the single low-pass subband (LL), at the bottom level. Up to 6 resolutions can be present in 2K, whereas 7 maximum / 2 minimum resolutions are allowed in 4K.
- *Subbands*: contain the horizontal and vertical low-pass/high-pass transform coefficients.

- *Precincts*: an optional tile of pixels at the resolution/subband levels. They are required, if used, to have power-of-two side sizes. The DCI standard specifies them to be used and to have a 128x128 size at the lowest resolution, and 256x256 size at the higher ones.[69]
- *Codeblocks*: a tile of pixels at the precinct level. They are required to have power-of-two side sizes. The JPEG 2000 standard specifies them to be this exact size everywhere, and the DCI standard specifies this size to be 32x32.[69]

Figure 54 shows how the BBB frame is fully partitioned after this step.

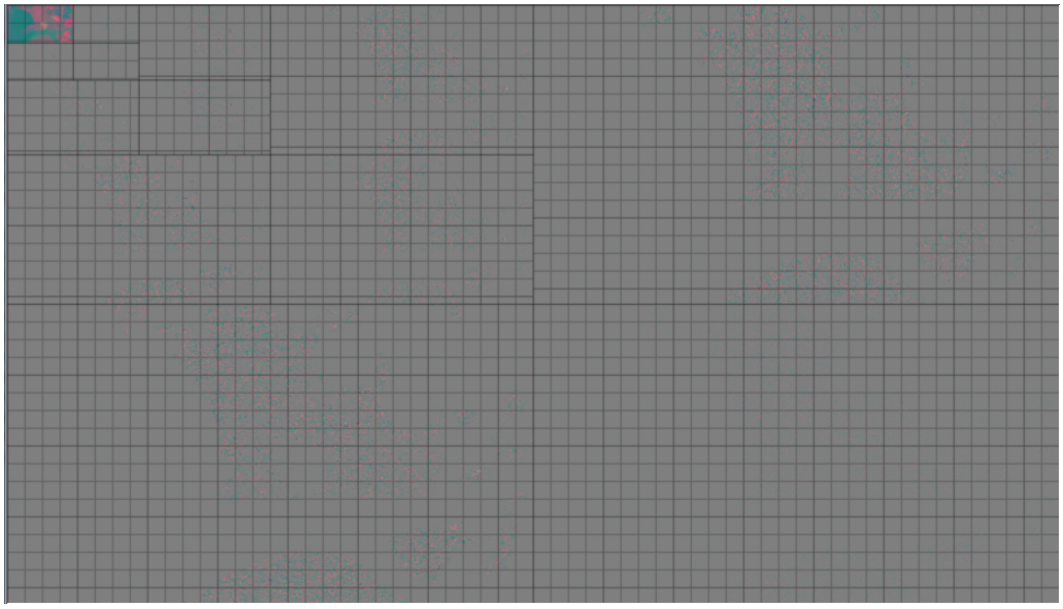


Figure 54 - The frame after sample data partitioning: every different solid line width represents a different partitioning level (subband, precinct, and codeblock).

Tier-1 Lossless coding

A second coding step requires now that compression is achieved in a lossless way, that is, the quantized coefficient samples are exactly coded into the compressed image.

Fractional bitplane coding

Every bitplane is scanned in a zig-zag fashion, and the bit values are arithmetically encoded depending on their surrounding "context", i.e., the values of other bits and how much is "significant" the sample that contains the bit.

[69] If the upper partitioning level size does not allow having exactly these sizes, they are cropped accordingly.

MQ arithmetic entropy coding

The arithmetic coder is used to lossless compress the context data and bit values, obtained during the fractional bitplane coding. This coder is adaptive, and can switch its operating context in response to changes in the bitplane contexts.

After coding, the weight of the codeblocks is shown for the BBB frame in fig. 55.

Figure 55 - Codeblock weight after arithmetic coding: lighter colors denote that more bits are used to represent that codeblock.

Tier-2 lossy coding and codestream formation

This third and last coding step reintroduces loss in reconstruction, but only to meet a specified bit rate/quality target. The compressed data are organized into a so called codestream, a particularly crafted sequence of bytes.

Layers and Post-Compression Rate-Distortion (PCRD) optimization

An internal representation of how compressed bits influence the reconstructed image quality is built, and the final bits are selected so as to maximize expected PSNR.

Bitstream packets and Tag Trees

Codeblock data are written in a way such as to allow progressive reconstruction during the decoding step. This is achieved via a particular compression structure called tag tree.

Codestream markers

The compressed data bitstream and headers are augmented with some general information on image size, depth, etc., as well as with decomposition parameters, and written into a codestream. If you want to see how a codestream is internally done, you can open a JPEG 2000 compressed image with *opjviewer*.[70]

Encoding performance of DCP Builder

The performance in terms of conversion time is not high: do not expect to achieve an encoding rate higher than 1.5-2 fps on a quad core machine. Just as an example, in table 7 are given some of the processing speeds of DCP Builder.

The test source is the Big Buck Bunny movie, 1920x1080 8 bit PNG pictures, 5.1 FLAC audio channels. The DCP is done in 24 fps, 48 kHz, SMPTE, 5.1 channels, HDTV format. The chosen colorspace is sRGB. The bit rate is set at 125 Mbit/s. The numbers are given for a single processing thread, and are expressed in milliseconds. The CPU is a Intel Quad Core i7 870 @ 3.0 GHz. The OS is Windows 7 64 bit. The optimizations in v02 have been done mainly using a cleaner C syntax and the SSE2 instruction set.[71] As you can well see, even after optimization, the heaviest processing is that of the JPEG 2000 compression, which ultimately limits the speed of the program.

[70] http://dsplab.diei.unipg.it/software/opjviewer_jpeg_2000_viewer

[71] SSE2 is available on any CPU (Intel or AMD) starting from Pentium IV. If you have an older CPU, SSE2 will not be used, but a slight improvement in processing time can be achieved as well.

Table 7 - Processing time for a single frame (ms).

Function	**Times in v0.1**	**Times in v0.2**
Picture load	0	120-150
Gamma decoding	650-700	15-18
RGB→XYZ	95-110	18-20
Resize	COPY: 37-42	8-12
	BSPLINE: 450-500	450-500
Gamma encoding	640-670	15-20
J2K compression	2700-3800	2600-3700
Audio resampling	COPY: 0	0
	SINC: 15	15
Audio compression	0	0
MXF contention	30	30
Picture load	120-150	0

Project file syntax

By default, DCP Builder project files have the *.dcb* extension. They are text files that contain a basic syntax for specifying the actions to be done by the program. You can open a file and modify it with your preferred text editor.

The parameter names are case-sensitive; most of the parameter values are case-insensitive.

Comments

Lines beginning with '**#**' or '*;*' are considered comments. Empty lines are skipped.

Ex.:

```
# this is a comment

; this is a comment, too
```

Input video section

InputVideoFileName [string]

The file path and name of the input clip. Video content is mandatory, while audio content is optional (auxiliary file or silence, else).

Ex.:

```
InputVideoFileName = my_video.avi
```

InputLVideoFileName [string], InputRVideoFileName [string]

The file path and name of the input stereoscopic clips. Audio content is taken from aux or mono files.

Ex.:

```
InputLVideoFileName = sampleL.mov
InputRVideoFileName = sampleR.mov
```

InputVideoFrameRate [float]

The frame rate of the input file(s), if not recoverable from the file itself.

Ex.:

```
InputVideoFrameRate = 24
```

InputVideoDownSample [integer]

Frame increment to use: 1 means every frame.

> NOTE: THIS MAY NOT WORK AS EXPECTED. DON'T USE UNLESS NECESSARY.

Ex.:

```
InputVideoDownSample = 2
```

InputVideoRepeat [integer]

Number of times the whole clip will be repeated.

> NOTE: THIS DOES NOT WORK YET.

Ex.:

```
InputVideoRepeat = 1
```

InputVideoColourSpace [string]

Colorspace of the input video: *XYZ*, *BT.709*, *sRGB*, *PAL*, *NTSC*, *Adobe*, *Apple*, *Custom*.

Ex.:

```
InputVideoColourSpace = PAL
```

InputVideoCustomTransform [list of 9 floats]

The 9 custom color transform matrix coefficients.

Ex.:

```
InputVideoCustomTransform = 0.26 0.35 0.12 0.02 0.00 0.78 0.21 0.36 0.10
```

InputVideoGamma [float]

Gamma of the input video, e.g. 2.2. You may here override the default colorspace gamma.

Ex.:

```
InputVideoGamma = 2.35
```

Input audio section

InputAudioFileName [string]

The file path and name of an auxiliary multichannel file from where audio can be optionally taken, if the main file has not it.

Ex.:

```
InputAudioFileName = mysound.wav
```

InputXAudioFileName [string]

The file path and name of auxiliary mono files from where audio can be optionally taken, if the main file has not it. Up to 16 channels can be mapped.

Ex.:

```
InputLAudioFileName    = onechannel01.wav
InputRAudioFileName    = onechannel02.wav
InputCAudioFileName    = onechannel03.wav
InputLFEAudioFileName  = onechannel04.wav
InputRSAudioFileName   = onechannel05.wav
InputLSAudioFileName   = onechannel06.wav
InputLCAudioFileName   = onechannel07.wav
InputRCAudioFileName   = onechannel08.wav
Input09AudioFileName   = onechannel09.wav
Input10AudioFileName   = onechannel10.wav
Input11AudioFileName   = onechannel11.wav
Input12AudioFileName   = onechannel12.wav
Input13AudioFileName   = onechannel13.wav
Input14AudioFileName   = onechannel14.wav
Input15AudioFileName   = onechannel15.wav
Input16AudioFileName   = onechannel16.wav
```

InputAudioLayout [float]

Channel layout: *1.0*,[72] *2.0*, *5.1*, *7.1*, or *16*.

Ex.:

```
InputAudioLayout = 2.0
```

Input text files

InputTextFileName [string]

The file path and name of a *.srt* text subtitle file.

Ex.:

```
InputTextFileName = my_subtitles.srt
```

DCDM section

DCDMPicturePath [string]

Folder path where TIFF files are optionally stored.

[72] 1.0 is only supported by SMPTE packages.

> NOTE: *.PPM* FILES ARE STORED, INDEED.

Ex.:

```
DCDMPicturePath = dcp/dcdm/tiff
```

DCDMXmlPath [string]

Folder path where Xml files are optionally stored.

Ex.:

```
DCDMXmlPath = dcp/dcdm/xml
```

DCDMSoundPath [string]

Folder path where Wave files are optionally stored.

> NOTE: *.AU* FILES ARE STORED, INDEED.

Ex.:

```
DCDMSoundPath = dcp/dcdm/wav
```

DCDMSaveType [string]

Type of DCDM to save. Space separated list of: *tiff*, *j2k*, *wave*, *xml*.

Ex.:

```
DCDMSaveType = tiff wave
```

DCDMPictureResolution [string]

Frame size of the DCDM: *2K*, *4K*, or *3D*.

Ex.:

```
DCDMPictureResolution = 2K
```

DCDMPictureResolutionLevels [string]

Number of embedded resolutions in the DCP, may vary between 1 and 7 (the maximum value is dictated by the picture resolution).

Ex.:

```
DCDMPictureResolutionLevels = 5
```

DCDMPictureRate [integer]

Frame rate of the DCDM: *23.98*, *24*, *25*, *30*, *48*, *50* or *60* fps.

Ex.:

```
DCDMPictureRate = 24
```

DCDMSoundRate [integer]

Sampling rate of the DCDM: *48000* or *96000* Hz.

Ex.:

```
DCDMSoundRate = 48000
```

DCDMPictureAspectRatio [string]

Aspect ratio of the DCDM: *flat*, *scope*, *hdtv*, *full*, or *custom*. Custom ratio is given in **DCDMPictureCustomAspectRatio**.

Ex.:

```
DCDMPictureAspectRatio = scope
```

DCDMPictureCustomAspectRatio [string]

Custom aspect ratio of the DCDM, given as *widthxheight* (e.g., *2048x800*). Must be enabled in **DCDMPictureAspectRatio**.

Ex.:

```
DCDMPictureCustomAspectRatio = 2048x872
```

> NOTE: USING A CUSTOM ASPECT RATIO MAY RENDER THE DCP UNPLAYABLE ON CERTAIN SERVERS.

DCDMPictureBoxing [string]

Boxing type of DCDM: *pillarbox*, *letterbox*, or *stretch*.

> NOTE: BOXING MAY NOT APPEAR, DUE TO RELATIVE SIZES.

Ex.:

```
DCDMPictureBoxing = letterbox
```

DCDMSubtitleMethod [string]

Subtitles are added in several ways: *none*, *XML+text*, *XML+PNG*, *Burn-in*. The support of subtitles is not guaranteed, please check the log for further details.

Ex.:

```
DCDMSubtitleMethod = XML+text
```

DCDMJ2KPicturePath [string]

Folder path where JPEG 2000 codestreams are stored, optionally.

Ex.:

```
DCDMJ2KPicturePath = dcp/dcdm/j2k
```

DCDMNumberOfFrames [integer]

Define the number of frames to store in the DCP. Set to 0 for the complete length of the input clip.

Ex.:

```
DCDMNumberOfFrames = 0
```

DCDMFramesPerReel [integer]

Limit of frames to assign to each single reel: 1550 is a maximum of 2GB (good for FAT32 disks). A number of 0 (manual mode) means that the cue list will be used. In case the cue list is empty, only one reel will be created.

Ex.:

```
DCDMFramesPerReel = 10
```

DCDMCueList [string]

The cue points are used to split the DCP in reels. 0 is not a valid cue point. Points can be expressed as a space/tab/comma separated list of frame numbers. This list is used only if **DCDMFramesPerReel** is set to manual (0).

Ex.:

```
DCDMCueList = 2560 15231 18590
```

DCDMPictureBitrate [float]

Maximum bit rate, in bit/s, of the total picture part of the DCDM*.

Ex.:

```
DCDMPictureBitrate = 239000000
```

DCDMBitrate2 [float]

Maximum bit rate, in bit/s, of the per component picture part of the DCDM*.

Ex.:

```
DCDMPictureBitrate2 = 185000000
```

DCDMBypass [string]

The type of bypass to apply on the DCDM: a combination of *gamma*, *headroom*.

Ex.:

```
DCDMBypass = gamma
```

DCP section

DCPPath [string]

Folder path where the DCP files are stored. It will be created if it does not exist.

Ex.:

```
DCPPath = dcp/DCI_checks
```

DCPType [string]

Type of movie: one of *feature*, *trailer*, *test*, *teaser*, *rating*, *advertisement*, *short*, *transitional*, *psa*, *policy*.

Ex.:

```
DCPType = test
```

DCPVersion [integer]

Numeral of the version of the DCP created.

Ex.:

```
DCPVersion = 1
```

DCPClipTitle [string]

Title of the clip contained in the DCP.

Ex.:

```
DCPClipTitle = Big Buck Bunny
```

DCPTitle [string]

Title of the DCP.

Ex.:

```
DCPTitle = BBB_SHR_C_EN-XX_INT-G_51-EN_2K_DSPL_20110214_UPG_OV
```

DCPAnnotation [string]

Annotation text of the DCP.

Ex.:

```
DCPAnnotation = BBB_SHR_C_EN-XX_INT-G_51-EN_2K_DSPL_20110214_UPG_OV
```

DCPIssuer [string]

Issuer text of the DCP.

Ex.:

```
DCPIssuer = DSPLab
```

DCPFacility [string]

Facility text of the DCP.

Ex.:

```
DCPFacility = UPG
```

DCPAgency [string]

Rating agency text of the DCP.

Ex.:

```
DCPAgency = Unipg
```

DCPLabel [string]

Rating label of the DCP.

Ex.:

```
DCPLabel = G
```

DCPLocale [string]

Use two or three letter ISO codes for languages and countries, as specified.[73] These parameters do not affect directly the DCP itself, they are used by the GUI to suggest a DCP title following to DC naming convention.

Ex.:

```
DCPLocaleAudio = EN
DCPLocaleSubtitles = XX
DCPLocaleNarrative = ES
DCPLocaleTerritory = INT
```

DCPStandard [string]

DCP compatibility standard: *Interop* or *SMPTE*.

Ex.:

```
DCPStandard = smpte
```

Names of the DCP files [string]

The XML and MXF files can be automatically named.

- The term *$U* expands to the UUID of the relevant asset.
- The term *$R* expands to a new random UUID.
- C language *printf()* format parameters can be used for automatic numbering of MXF reels and XML files.
- Asset map and volume index must follow the guidelines dictated by the standards.

Ex.: in this case, "*%03d*" expands to "*000*", "*001*", "*002*", and so on, at every new reel.

```
DCPAssetMapName             = ASSETMAP.xml
DCPVolumeIndexName          = VOLINDEX.xml
DCPCompositionPlaylistName  = cpl.xml
DCPPackingListName          = pkl.xml
DCPPictureReelName          = videoreel%03d.mxf
DCPAudioReelName            = audioreel%03d.mxf
DCPTextReelName             = textreel%03d.mxf
```

Ex.: in this case "*$U*" expands to something like "*7e2b76ac-43ff-406c-9e74-ee34e04d3f79*", which is that asset's UUID, and it is different at every new reel.

[73] http://www.digitalcinemanamingconvention.com/

```
DCPAssetMapName             = ASSETMAP
DCPVolumeIndexName          = VOLINDEX
DCPCompositionPlaylistName = $U_cpl.xml
DCPPackingListName          = $U_pkl.xml
DCPPictureReelName          = $U_picture.mxf
DCPAudioReelName            = $U_sound.mxf
DCPTextReelName             = subtitle.mxf
DCPFontName                 = font.ttf
```

Cryptography section (EXPERIMENTAL - DON'T USE)

CryptoEncrypt [boolean]

Encrypt picture, sound, and text (only SMPTE) reels using recommended AES-128 cypher in CBC mode. The resulting DCP can only be played on authorized servers and in authorized time slots, via a KDM.

Ex.:

```
CryptoEncrypt = yes
```

Compression engine section

EngineNumberOfThreads [integer]

Number of concurrent threads to use. Adapted to CPU number: *0*.

Ex.:

```
EngineNumberOfThreads = 1
```

EnginePictureResize [string]

Video resizing method. From worst to best: *copy*, *nearest*, *bilinear*, *lanczos*, *bicubic*, *bspline*.

Ex.:

```
EnginePictureResize = bspline
```

EngineSoundResample [string]

Audio resampling method. From worst to best: *copy*, *nearest*, *linear*, *sinc*.

Ex.:

```
EngineSoundResample = sinc
```

EngineResume [string]

The program will try to resume encoding from where it left (*none*, *dcdm*, or *dcp*).

> NOTE: DCDM RESUME METHOD ONLY TAKES J2K FILES (I.E., DCDM*).
>
> NOTE: DCP RESUME METHOD DOES NOT WORK.

Ex.:

```
EngineResume = none
```

EngineTraceFileName [string]

The file name of a text file that will contain the encoding trace.

Ex.:

```
EngineTraceFileName = trace.txt
```

EngineTraceContents [list of strings]

The type of encoding trace. May contain a combination of *none*, *psnr, bitrate*, and *fps*.

Ex.:

```
EngineTraceContents = psnr bitrate fps
```

EngineLogFileName [string]

The file name of a text file that will contain the encoding log.

Ex.:

```
EngineLogFileName = log.txt
```

EngineLog [list of strings]

The type of encoding log. May contain a combination of *none*, *all*.

Ex.:

```
EngineLog = all
```

Registration

By default, all the DCPs done with DCP Builder are watermarked with the name of the program in every frame (sort of paranoid, I know, see later “Why I ask for your machine fingerprint?”).

In order to remove this watermark, you need to obtain a valid registration code. The program is able to generate a *machine fingerprint* that uniquely identifies your machine, based on certain HW characteristics. The machine fingerprint and the registration code are paired in such a way that, if correct, the watermark is disabled. The fingerprint itself is an alphanumeric code composed by ten groups of four capital letters in the range 0-9 and A-F, each group separated by a dash sign.

Sometimes it can happen that your fingerprint changes. This is mostly due to slight changes in the hardware configuration of your PC. For example, activating/deactivating a wireless card, inserting/removing a USB disk, etc. It is thus suggested to take note of the fingerprint during several days of operation, and to choose the one recurring more often.

You can obtain this machine fingerprint via the ***Help→Registration*** menu or by hitting ***F3***. The shown dialog (fig. 56) presents the fingerprint (replicated also in the log panel, so you can copy-paste it) and a field where you can enter the corresponding registration code.

Enter registration code
The created DCPs are all watermarked with a logo, which can be removed with a registration code. In order to obtain a code, you must follow the instructions provided by the author, which you find here
http://dsplab.diei.unipg.it/software/dcp_builder
Your machine fingerprint is
(you can copy this serial from the log panel also)
Enter the received registration code:
REGISTRATIONCODE
OK
Cancel

Figure 56 - Dialog used to find the machine fingerprint and enter the registration code.

Before sending your fingerprint, restart the machine several times under different configurations (wired network, Wi-Fi dongle, external USB drives, etc.) and record the fingerprint. It should not change, but in case it changes, please send the one that appears more frequently.

You should not install DCP Builder on your laptop. Laptops are not expected to be used to create DCPs: they have slow CPUs, slow RAM, slow disks, etc. Avoid sending the fingerprint of your laptop, even for evaluation purposes: if you need to evaluate, you can well use the watermarked version, then decide on which machine to install for production, and eventually request a registration code. Moreover, laptops suffer from the changing fingerprint issue, so you are warned.

Remember that the registration code is linked to the fingerprint, which is UNIQUE for every PC. Thus, only request a code with the fingerprint of the PC on which you will prepare the DCPs.

NOTE: DCP PREPARATION FACILITIES CAN USE THIS PROGRAM, PROVIDED THAT A SEPARATE AGREEMENT IS DONE.

How to obtain a registration code

In order to obtain a registration code please refer to the instructions provided in the official websites

http://dsplab.diei.unipg.it/software/dcp_builder

http://www.dcpbuilder.com/

Why I ask for your machine fingerprint?

One of the main reasons is that it would be possible for others to embed this program in a GUI or CLI of their own and claim it as their own, possibly even requiring to be paid for this. The fingerprint/registration code should prevent this to the less skilled pirates.

I do not collect your data neither for sending spam or bothering you with various requests - they will remain private - nor for claiming money.

Frequently Asked Questions (FAQ)

1) I need this program, but don't really understand why you want all my personal information details.

At the beginning of this manual there is an updated list of programs that can do DCPs, both free and commercial; you are not obliged to use DCP Builder.

2) I want to test the program, please send me the registration code

If you just need to test the program, there is no need to ask a registration code. The DCP will be watermarked in the center of every frame but, apart from that, it will work perfectly as if registered. After you have tested it successfully and you are satisfied with it, then apply for your registration.

3) I am not employed by any company. Can I get the code as well?

Of course, that is possible. Just specify that in your request - in addition to all other required fields.

4) Why the "one code per employer" limit?

This program is not conceived to be used as a DCP mass production tool. Large companies are well able to buy commercial licenses of other similar tools.

5) I have asked for a registration code, but did not receive it. How do I get it?

Please be sure to fill all the required fields in the email and allow some days for processing. No answer (positive or negative) could be given to badly compiled or lacking in detail emails. Moreover, be explicit in the title of the email, such as "Request of registration code for DCP Builder".

6) Encoding is slow. How can I improve speed?

Every effort has been done to make things go fast. Faster encoding needs costly libraries, contact the author for possible solutions. If you are strongly interested in faster DCP encoding and can fund development studies, just try to ask.

7) When I start the program, all commands seem to be in a frozen state..

That is normal. You just need to start a new project (empty, prebuilt, or simple wizard) and then it will get defrost.

8) The program starts encoding and after a variable amount of time, it crashes.

Are you running a 32 bit OS? Try to lower the number of threads (e.g., 2). Unfortunately, the OpenJPEG compression engine requires a lot of memory. A 32 bit OS (or 32 bit subsystems under a 64 bit OS) can only allocate a maximum amount of 2 GB of RAM to ALL its 32 bit applications. From my experience, close any other program you have open (browsers are especially memory-eager) and avoid exceeding 1.2 GBytes of used memory for the tool.

9) 4K encoding is very slow and crashes the program.

Crashing happens generally when too much memory is being requested by the application, and you are running under a 32 bit OS. Try to set the number of threads to 1, before starting, and, when encoding, increase it up to an acceptable limit. Differently, run a 64 bit native version of the program (this is different from running a 32 bit version on a 64 bit OS).

10) What happens when input clip and DCP frame rates are different?

Generally speaking, nothing bad, if they are not too different. The program will always try to maintain the audio in sync with the video, thus, for example, a 25 fps input clip converted into a 24 fps DCP will result in audio slow-down. No processing is done to adjust the pitch, you get what you get. If your concern is on the audio quality, you should adjust the clip before converting it into a DCP.

11) I want the supercalifragilisticexpialidocious feature on the program. When will you add it?

Just try to ask for it, ONCE, and wait (it may be a very long wait).

12) Can I use the program to prepare a DCP for distribution to cinemas?

Of course, yes. You can also consider the “License” section, that is, you may want to credit this software and not try to sue me if it does not work as expected. In case, a free entrance for the premiere is welcome!

13) I want/need the source code? How do I get it?

Source code is not available. The only source code you can get is that of Libav, that for legal and licensing reasons must be made available to everyone. You can download it from

http://dante.diei.unipg.it/~baruffa/DCPBuilder/

14) I am not able to run the program. Can you prepare the DCP for me? Can you give me the best settings?

No, it is not possible. Please refer to a DC encoding facility or service, or experiment by yourself. Please don't ask things that are already covered in this manual.

15) Why did you choose Libav instead of FFmpeg?

I have no preference, for the moment I will stay on Libav, waiting to understand how the FFmpeg developer team will respond to the fork.

16) Why don't you provide ProRes support by default?

ProRes can be decoded only under Windows if you previously installed the QuickTime environment. The QuickTime SDK is required to build the program with this support enabled. It is uncertain what exactly are the licensing terms of such SDK, that's the motivation.

17) Do I need to select XYZ colorspace and 2.6 gamma, as per DCI specs?

Generally, you don't, unless your source material has been formatted in this way. Normally you have an input that is in sRGB or in BT.709 colorspace, try to select them.

18) Is it better to export my input sources to TIFF or to DPX?

For me, it's better TIFF. But others might prefer a different format. However, always try to use your original clip. If it's an AVI, why exporting to TIFF?

19) Why there is no more AVI and MOV support?

In order to avoid risks of patent infringement (e.g., H.264, MPEG-2), all the FFmpeg or Libav code used for this, has been removed. This has nothing to do with the LGPL licensing, which was completely fulfilled. You can personally enable support to AVI and MOV (and other types of video as well), however.

20) How can I add further presets in the wizard?

Presets are partially completed .dcb projects which are placed in a default directory in your system. This folder is generally in the installation directory: c:\programs\dcpbuilder\template under windows, in /usr/share/dcpbuilder/template (or similar, use find for the exact location) under linux, or inside the Contents/MacOS/template app folder under mac os x. So, basically you just need to create a DCP builder project file with all your preferred settings, save it with a significant name (for you), and place it in the template folder. When you restart the program, it will find the new template, that will be selectable in the ***File→new*** menu.

License

DCP Builder

Basically:

- you can download DCP Builder and install it on whatever number of PCs you want,
- you can run an unlimited number of instances of DCP Builder, in parallel or concurrently, on whatever PC you use,
- you can produce an unlimited number of DCPs, either for personal, non-profit, or commercial use,
- you can receive only ONE registration code per employer,[74]

provided that (the following is not mandatory, but any elegant person should consider it)

- you clearly show in your derived work (e.g., final credits) that this software has been used for DCP production, the web page you downloaded it from, and the copyright holder statement, for instance with the following credit lines:

  "**This DCP has been produced using DCP Builder 0.3,**
  Copyright © 2009-2012 Giuseppe Baruffa, All Rights Reserved,
  http://dsplab.diei.unipg.it/software/dcp_builder
  http://www.dcpbuilder.com/"

Anyway, I will not sue you if you don't![75] But, remember that writing this program took me a lot of time and efforts, and that you are not paying a buck for it, so just be honest.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

[74] This means that there is no 100% guarantee that you will receive at least one free registration code, the number may be between 0 and *n*, it depends on the discretion of the author.

[75] For example, if you use DCP Builder for 10 s interval clips or for ads, you can safely omit this stuff.

DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Included libraries

Zlib

Copyright (C) 1995-2010 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly

Mark Adler

Libbzip2

This program, "bzip2", the associated library "libbzip2", and all documentation, are copyright (C) 1996-2007 Julian R Seward. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, jseward@bzip.org

bzip2/libbzip2 version 1.0.5 of 10 December 2007

LibTIFF

Copyright (c) 1988-1997 Sam Leffler

Copyright (c) 1991-1997 Silicon Graphics, Inc.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that (i) the above copyright notices and this permission notice appear in all copies of the software and related documentation, and (ii) the names of Sam Leffler and Silicon Graphics may not be used in any advertising or publicity relating to the software without the specific, prior written permission of Sam Leffler and Silicon Graphics.

THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Libav

This software library adopts the *GNU Lesser General Public License, version 2.1*, which you can find in the install folder of the program, also:

http://www.gnu.org/licenses/old-licenses/lgpl-2.1-standalone.html

LibPNG

This software library adopts an open source license:

http://www.libpng.org/pub/png/src/libpng-LICENSE.txt

Libnsbmp

This software library adopts the MIT license.

http://opensource.org/licenses/mit-license.php

LibFLAC

This software library adopts the new BSD license.

http://flac.cvs.sourceforge.net/viewvc/flac/flac/COPYING.Xiph

LibTarga

This software library adopts the MIT license.

http://opensource.org/licenses/mit-license.php

Libxml2

This software library adopts the MIT license.

http://opensource.org/licenses/mit-license.php

OpenDPX

This software library adopts the new BSD license.

http://www.opensource.org/licenses/bsd-license.php

OpenJPEG

This software library adopts the new BSD license.

http://www.opensource.org/licenses/bsd-license.php

SQLCipher

Copyright (c) 2010 Zetetic LLC

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the ZETETIC LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY ZETETIC LLC "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ZETETIC LLC BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

XMLSec

This software library adopts the MIT license.

http://opensource.org/licenses/mit-license.php

Bitstream Vera Sans

Copyright (c) 2003 by Bitstream, Inc. All Rights Reserved. Bitstream Vera is a trademark of Bitstream, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of the fonts accompanying this license (“Fonts”) and associated documentation files (the “Font Software”), to reproduce and distribute the Font Software, including without limitation the rights to use, copy, merge, publish, distribute, and/or sell copies of the Font Software, and to permit persons to whom the Font Software is furnished to do so, subject to the following conditions:

The above copyright and trademark notices and this permission notice shall be included in all copies of one or more of the Font Software typefaces.

The Font Software may be modified, altered, or added to, and in particular the designs of glyphs or characters in the Fonts may be modified and additional glyphs or characters may be added to the Fonts, only if the fonts are renamed to names not containing either the words “Bitstream” or the word “Vera”.

This License becomes null and void to the extent applicable to Fonts or Font Software that has been modified and is distributed under the “Bitstream Vera” names.

The Font Software may be sold as part of a larger software package but no copy of one or more of the Font Software typefaces may be sold by itself.

THE FONT SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF COPYRIGHT, PATENT, TRADEMARK, OR OTHER RIGHT. IN NO EVENT SHALL BITSTREAM OR THE GNOME FOUNDATION BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, INCLUDING ANY GENERAL, SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF THE USE OR INABILITY TO USE THE FONT SOFTWARE OR FROM OTHER DEALINGS IN THE FONT SOFTWARE.

Except as contained in this notice, the names of Gnome, the Gnome Foundation, and Bitstream Inc., shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Font Software without prior written authorization from the Gnome Foundation or Bitstream Inc., respectively. For further information, contact: fonts at gnome dot org.

Glossary

This is a short list of terms and abbreviations used in this document.

Acronym	Meaning
2K	2048×1080 screen resolution
4K	4096×2160 screen resolution
3D	Three-dimensional cinema
AVI	Audio Video Interleaved file format
BBB	Big Buck Bunny
CPL	Composition PlayList
DC	Digital Cinema
DCDM	Digital Cinema Distribution Master
DCDM*	Compressed Digital Cinema Distribution Master
DCP	Digital Cinema Package
ext2, ext3	Second, third extended file system
FAT32	File Allocation Table, 32 bit file system
FPS	Frames Per Second
HDTV	High Definition Television, 1920×1080 screen resolution
ICT	Irreversible Color Transform
JPEG	Joint Picture Experts Group
KDM	Key Delivery Message
MOV	QuickTime® clips file name extension
MXF	Material eXchange Format
PCM	Pulse Code Modulation
PCRD	Post-Compression Rate-Distortion
PKL	PacKing List

Acronym	Meaning
PSNR	Peak Signal-to-Noise Ratio
SAR	Screen Aspect Ratio
SRT	SubRip Text
TIFF	Tagged Image File Format
XML	eXtensible Markup Language
XYZ	CIE 1931 color space

Bibliography

[1] Glenn Kennel. *Color and Mastering for Digital Cinema*. Focal Press, 2006.

[2] Rafael Gonzalez, Richard Woods. *Digital Image Processing*. Prentice Hall, 2008.

[3] Henry R. Kang. *Computational colour technology*. SPIE Press, Bellingham, WA, USA, 2006.

[4] Al Bovik (editor). *Handbook of image and video processing*. Academic Press, 2000.

[5] David Taubman, Michael Marcellin. *JPEG2000: Image Compression Fundamentals, Standards and Practice*. The Kluwer International Series in Engineering and Computer Science, Kluwer Academic Publishers, 2004.

[6] Tinku Acharya, Ping-Sing Tsai. *JPEG2000 standard for image compression: concepts, algorithms and VLSI architectures*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2005.

[7] James H. Kasner. *Bandwidth Compression (BWC) Guide for JPEG 2000 Visually Lossless and Numerically Lossless Compression of Imagery Data - Working Draft 1.2*. Eastman Kodak Company, January 22nd, 2004.

[8] Xavier Delaunay. *EBCOT coding passes explained on a detailed example*. [Online]. Available: http://d.xav.free.fr/ebcot/EBCOT_example.pdf.

[9] Gaetano Impoco. *JPEG2000 - A Short Tutorial*. April 1st, 2004. [Online]. Available: http://www.dmi.unict.it/~impoco/files/tutorial_JPEG2000.pdf.

[10] Jin Li. *Image Compression - the Mathematics of JPEG 2000*. Microsoft Research, Communication Collaboration and Signal Processing, 2002. [Online]. Available: http://research.microsoft.com/en-us/um/people/jinl/paper_2002/msri_jpeg.pdf.

[11] *Digital Cinema System Specification*, Digital Cinema Initiatives, LLC Std. V1.1, 2007. [Online]. Available: http://www.dcimovies.com.

[12] *Stereoscopic Digital Cinema Addendum*, Digital Cinema Initiatives, LLC Std. V1.0, 2007. [Online]. Available: http://www.dcimovies.com.

[13] ISO/IEC Standard. *ISO/IEC 15444-1:2004. Information technology -- JPEG 2000 image coding system - Part 1: Core coding system*. 2004.

[14] ISO/IEC Standard. *ISO/IEC 15444-1:2004/Amd 1:2006. Information technology – JPEG 2000 image coding: Core coding system. AMENDMENT 1: Profiles for digital cinema applications*. 2006.

[15] Stan Cossette. *MPEG Interoperability Initiative: Composition Playlist Specification - Application Specification for Digital Cinema Packaging (AS-DCP)*. V. 2.4, July 8th, 2004.

[16] Stan Cossette. *MPEG Interoperability Initiative: Delivery Media Representation and Segmentation - Application Specification for Digital Cinema Packaging (AS-DCP)*. V. 3.4, August 24th, 2004.

[17] Stan Cossette. *MPEG Interoperability Initiative: Packing List Specification - Application Specification for Digital Cinema Packaging (AS-DCP)*. V. 2.4, May 27th, 2005.

[18] Tim Ryan. *Subtitle Specification (XML File Format) for DLP Cinema™ Projection Technology*. Rev. B, May 12th, 2003.

[19] SMPTE Standard. *D-Cinema Packaging — Composition Playlist, SMPTE 429-7-2006*. October 3rd, 2006.

[20] SMPTE Standard. *D-Cinema Packaging — Packing Playlist, SMPTE 429-8-2007*. June 25th, 2007.

[21] SMPTE Standard. *D-Cinema Packaging — Asset Mapping and File Segmentation, SMPTE 429-9-2007*. June 25th, 2007.

[22] SMPTE Standard. *Digital Cinema Distribution Master — Subtitle, SMPTE 428-7-2007*. April 12th, 2007.

[23] SMPTE Standard. *D-Cinema Distribution Master — Audio Characteristics, SMPTE 428-2-2006*. August 16th, 2006.

[24] SMPTE Standard. *D-Cinema Distribution Master — Image Characteristics, SMPTE 428-1-2006*. September 29th, 2006.

[25] Wikimedia Foundation Inc. *Wikipedia: The Free Encyclopedia*. Updated 10 May 2011, 18:00 UTC. Encyclopedia on-line. Available from http://en.wikipedia.org/. Internet. Retrieved 10 May 2011.

Credits

People

The main author of the program is Giuseppe Baruffa,[76] assistant professor at the Department of Electronic and Information Engineering[77] of the University of Perugia,[78] and member of the Digital Signal Processing laboratory. It took several time to write the base library, the command line tool, and the graphical interface.

Here the acknowledgements come.

First of all, Pietro Clarici, theatre owner of Supercinema and Politeama Clarici, Foligno,[79] without whom I could have not tested and projected the GBytes of data produced by this software.

Then, Wolfgang Woehl, Filmmuseum Munich,[80] who suggested some neat improvements on the interface and on sane default behavior.

Knut Erik Evensen[81] has tested the Mac OS X port and helped to fill the bundle application with all correct shared libraries. He also tested the QuickTime version with Apple Prores clips.

Antonin Descampe, Université catholique de Louvain, has reported a bug in Mac OS X interface.

Bill Elswik and Michael Karagosian[82] for providing access to MPEG Interoperability Initiative documents.

Kommer Kleijn has fixed some errors and misconceptions written into this manual.

David Ballard tested the program on an extremely multithreaded machine, with 12 cores (24 threads).

And a special thank goes to the boss and the people of the DSP Laboratory: Fabrizio, Paolo, and Barbara.

---

[76] http://dsplab.diei.unipg.it/people/faculty/giuseppe_baruffa.

[77] http://www.diei.unipg.it/.

[78] http://www.unipg.it/ .

[79] http://folignoalcinema.wordpress.com/.

[80] http://github.com/wolfgangw/digital_cinema_tools/wiki.

[81] http://www.knuterikevensen.com/.

[82] http://www.mkpe.com/.

Auxiliary libraries, software, and other things

DCP Builder uses the following external or internal libraries (version numbers may change, check the log panel right after startup or hit ***F2*** to show the credits dialog for exact numbers):

- zlib v. 1.2.3.3 - (C) 1995-2004 Jean-loup Gailly and Mark Adler.
- bzip2 v. 1.0.5, 10-Dec-2007 - (C) 1996-2002 Julian R Seward. All rights res.
- OpenJPEG v. 1.4.0 - (C) 2002-2007, Communications and Remote Sensing Laboratory, Universite catholique de Louvain (UCL), Belgium.
- LIBTIFF, Version 3.8.2 - Copyright (c) 1988-1996 Sam Leffler, Copyright (c) 1991-1996 Silicon Graphics, Inc.
- libpng version 1.2.15beta5 - December 8, 2006 (header) . Copyright (c) 1998-2010 Glenn Randers-Pehrson.
- FreeType v. 2.3.5 - (C) 2003, 2006, 2007 The FreeType Development Team (www.freetype.org).
- Bitstream Vera Sans - Copyright (c) 2003 by Bitstream, Inc. All Rights Reserved. Bitstream Vera is a trademark of Bitstream, Inc.
- AS-DCP v. 1.8.44 - Copyright (c) 2003-2011, John Hurst, All rights reserved.
- OpenCinemaTools v. 1.1.2 - Copyright (c) 2008 TELE-UCL, Benoit Macq, Marc Vandenbosch(MVA).
- OpenSSL 0.9.8g 19 Oct 2007 - Copyright (c) 1998-2003 The OpenSSL Project. All rights reserved.
- Expat 2.0.1, Copyright (c) 1998-2000 Thai Open Source Software Center Ltd and Clark Cooper, Copyright (c) 2001-2006 Expat maintainers.
- CIE tables - Copyright (c) 1995-2010 Color and Vision Research Labs.
- RGB spaces - Copyright (c) 2001-2010 Bruce J. Lindbloom. All Rights Reserved.
- The SGI Image File Format - Draft version 0.97, Paul Haeberli - paul@sgi.com - Silicon Graphics Computer Systems.
- Simplified TARGA library - Copyright (c) 2005 Michael L. Gleicher.
- OpenDPX v0.5.0 (SMPTE V2.0) - Copyright (c) 2009, Patrick A. Palmer. All rights reserved.
- NetSurf's libnsbmp v.0.0.2, Copyright 2006 Richard Wilson <richard.wilson@netsurf-browser.org>, Copyright 2008 Sean Fox <dyntryx@gmail.com>
- wxWidgets 2.9 - Copyright (C) 1997 Julian Smart, Markus Holzem.

- wxMathplot 0.1.2 - Copyright (c) David Schalig, Davide Rondini
- LibOGG 1.1.4 (C) Copyright 1994-2007 by the Xiph.Org Foundation http://www.xiph.org/
- libFLAC 1.2.1 - Free Lossless Audio Codec library, Copyright (C) 2000-2007 Josh Coalson
- Base64 coder - Copyright (c) 2001 Bob Trower, Trantor Standard Systems Inc.
- SQLCipher v3.7.9, Copyright (c) 2010 Zetetic LLC

Changelog

HISTORY OF MODIFICATIONS AND BUG FIXING

2012-02-03, DCP Builder Basic Edition 0.3
* Third version of the program.
* DPX files are now imported even with the "XPDS" signature.
* Removed automatic insertion of rating agency and rating label when they are not given
* Fixed last subtitle freeze bug in burn-in mode.
* Fixed simple wizard bad behaviors: two files are kept in stereoscopic mode, frame rate of input video is reflected in the input panel, and the rating tag naming convention issue.
* Completely removed internal support to AVI, MP4, MOV, etc. files via FFmpeg or Libav, in order to avoid patent infringement problems. This function must now be explicitly enabled by the final user.
* Moved position of <Creator> tag, and <ScreenAspectRatio> tag changed from rational to decimal, according to Interop XSD schemas (W. Woehl).
* Fixed a bug when no target DCP resolution is chosen.
* Added support to 96, 100, and 120 FPS in stereoscopic mode (avatar 2!), thanks to switch to AS-DCP 1.8.44.
* Renamed the current options to Basic Edition.

2011-07-18, DCP Builder v0.2
* Second version of the program.
* Added choice between a compact or extended GUI layout.
* Added full screen preview in real colors.
* Added new syntax elements: DCDMPictureResolutionLevels, DCPClipTitle, DCPBitrate2, DCPVersion.
* Added DCPCueList for manually-set reel interruption points.
* Moved a bunch of DCP parameters into the DCDM realm.
* Changed DCDMSize to DCDMResolution.
* Removed EngineCreateFolder from project files and added to program preferences.
* Added clip title field in DCP properties, it can be entered separately in the composition title.
* Added adjustable maximum bit rate of color components.
* Added selector for number of DWT resolutions.
* Added subtitle creation in Interop, SMPTE (experimental), and burn-in modes.

* Changed embedded font from Delicious to Bitstream Vera Sans, for copyright and licensing reasons.
* Added support to 25, 30, 50, and 60 fps (and relevant stereoscopic versions).
* Fixed access from multiple threads to the font burn-in process.
* Added a basic wizard to help novice users in creation of simple projects.
* Changed internal routine to read WAV files, instead of using Libav.
* Changed to libFLAC for reading FLAC files, instead of using Libav.
* Added email sending at the end of an encoding job.
* Added encoding log saving.
* Added CPU load and disk usage feedback in the GUI.
* Added switch between left and right views in preview for stereoscopy.
* Changed to new threading model.
* Removed support for JPEG inputs (it did not work, anyway).
* Added support for VOB inputs and anamorphic pixels.
* Changed gamma coding function internals, now 10x faster.
* Added multithreaded read of image sequences.
* Added chromatic adaptation check box in preferences dialog.
* Added DCDM creation bypass.
* Optimized speed in gamma conversion and colorspace conversion using even SSE2, if available.

2011-05-01, DCP Builder v0.1

* First version of the program.

Document history

Date	Author	Subject
2011-01-26	GB	First draft of instructions manual.
2011-02-12	GB	Clean up.
2011-03-11	GB	Updated figures and added some more clarifications.
2011-03-18	GB	Many small additions and refinements. Added a new parameter in project files.
2011-04-16	GB	New screenshots with better backgrounds. Better comparison table from en.wiki. New formats support.
2011-05-01	GB	Switching from FFmpeg to Libav. First public release.
2011-05-02	GB	Drafting version 0.2.
2011-07-15	GB	Final for version 0.2.
2011-08-23	GB	Drafting version 0.3. New style for bibliography.
2012-02-03	GB	Closing version 0.3.